Coded Modulation Systems Using General Block Codes: A Structured Distance Approach

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

by

ASHISH N JADHAV

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

October, 1997

Certificate

It is certified that the work contained in the thesis entitled CODED MODULATION SYSTEMS USING GENERAL BLOCK CODES: A STRUCTURED DISTANCE APPROACH, by Ashish N Jadhav, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

October, 1997

9.10.97

(M. U. Siddiqi)
Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

To
My Parents

Acknowledgments

My humble dedications,
TO GOD for creating "me",
TO MY PARENTS AND SISTER for teaching me to walk,
TO MY TEACHERS for showing me the way, and,
TO FAILURES for pushing me on.

I wish to express my deep sense of gratitude to my thesis supervisor, Dr. M. U. Siddiqi, for his guidance, advice, and continuous encouragement during the course of this work. I am indebted to him for the trust he reposed in me and patience during the difficult phases of this work. He is also responsible for carefully reading the first draft, making numerous corrections and at times saving me from blunders.

This work is a result of the countless sacrifices and unlimited patience of my parents. It is their faith and encouragement which enabled me to survive in the difficult phases of life.

My humble acknowledgments to Dr. V. P. Sinha, Dr. P. R. K. Rao and Dr. S. K. Mullick. They are the strong pillars on which the structure of the Electrical Engineering Department, which nurtured this work, rests. I consider my selves immensely lucky to get the opportunity to study under such dedicated teachers who have made an lasting impression on my personality and moulded my character. The motivating talks, discussions and helpful suggestions of Dr. V. P. Sinha have immensely contributed to my work.

But for the motivation and encouragement of Dr. K. V. Ramanan, Dr. N. S. T. Sai and A. C. Srivastava of R. A. I. T., I would have never enrolled for the Ph.D. program at I. I. T. Kanpur.

I thank Dr. P. K. Chatterjee, Dr. K. R. Srivathsan and Dr. S. Qureshi with whom I was associated as teaching assistant. Dr. P. Sircar, Dr. R. K. Bansal, Dr. S. Gupta and Dr. S. P. Mohanty taught some of the interesting courses. I also acknowledge Dr. P. Rao and Dr. S. Umesh for their helpful nature and for maintaining a congenial atmosphere in the lab.

It was the company of my seniors Dr. K. S. Venkatesh, G. Sinha and J. Mangasuli and their advice and words of encouragement which saw me through the times of despair. The presentation of this work in the present form would not have been possible without the prompt help provided by S. U. Khan and B. Singh.

Dr. Shamsiji, Pathakji and Setiaji maintained a jovial atmosphere during the hectic days of thesis completion. I am thankful to Dr. A. Chaturvedi, Dr. A. Karandikar, Dr. D. Murthy, Dr. Vinodkumar, Dr. K. Kapoor, Dr. Kanetkarji, J. Das, C. Ramesh, K. Chatterjee, M. Choudhary, P. Bhattacharya, A. Sharan, S. Dwivedi, S. Daniel, A. Harish, B. Singh, S. Upreti, P. Giri, U. Dikshit and to all my friends of Ghatmandal and E-mid/IV.

The facilities provided by the Central Library, Computer Center, Electrical Engineering Department and the IP-Lab were monumental in the smooth completion of this work.

I acknowledge Shri Attreji, Shri Bharatiaji, Shri Gopiji, Shri Tiwariji and Shri Vermaji, for creating a homely atmosphere in the campus.

Finally, I express acknowledgments to all those people known and unknown, who have helped me in whatsoever way during the thesis work.

Synopsis

The objective of the thesis is to develop a general framework for the study of **block coded modulation** [BCM] and present schemes for code search, block encoding and soft decoding of general (non-linear) block codes to be used in BCM systems over memory-less additive white Gaussian noise [AWGN] channels.

BCM is a coded modulation scheme which uses block codes. The major advantage of BCM is that general (non-linear) block codes which have a larger rate as compared to trellis coded modulation [TCM] [78] schemes, can be used with coded modulation.

Three types of signal constellations are referred to in the work:

(1) An **actual channel signal constellation** is the constellation whose signals are actually transmitted over the channel.

(2) An **virtual channel signal constellation** is a hypothetical signal constellation used in concatenated coded modulation schemes. It is, in fact, a block code which in turn uses an actual channel signal constellation.

(3) The general signal constellation which may be actual or virtual is termed as **arbitrary channel signal constellation**. An arbitrary channel signal constellation may be a PSK, ASK, QAM, or any other signal constellation in which the number of signals can be any number (not necessarily a prime or a power of a prime number).

A set-theoretic framework has been used for representation of signals from an arbitrary channel signal constellation and for sequences of signals of finite length. A block code of length n for a BCM scheme is defined to be a subset of the set $S' \times S' \times \ldots \times S'_{(n\text{-times})}$ of all sequences of length n of signals from an expanded channel signal constellation S'.

For working with various Euclidean distances, a matrix theoretic framework has been used. Euclidean distances between the signals of a channel signal constellation are represented in the form of a matrix. This matrix $\mathbf{d_S}$ is symmetric and its structure reflects the structural symmetries of the channel signal constellation. Such symmetries may be utilized

to reduce the storage requirements of Euclidean distances of signal constellations. The matrix $\mathbf{d_S}$ in turn is utilized to obtain a representation of the Euclidean distance between signal sequences. For a sequence of signals of length n from the finite set S' forming a channel signal constellation, the Euclidean distances between all the n-tuples are represented by the matrix

$$\mathbf{d}_{\mathrm{S \times S \times \ldots \times S (n\text{-}times)}} = \mathbf{d_S}(Ed)\mathbf{d_S} \ldots (Ed)\mathbf{d}_{\mathrm{S (n\text{-}times)}},$$

where (Ed) is a matrix operator which is structurally similar to the Kronecker product of matrices. The matrix $\mathbf{d}_{\mathrm{S \times S \times \ldots \times S (n\text{-}times)}}$ also provides the necessary information on the Euclidean distance distribution of sequences of signals of length n from the channel signal constellation.

For obtaining block codes the two basic approaches which have been used in the coding theory literature can be classified as follows:

(1) **The structured code approach:** In this approach, structure is imposed on the code words.

(2) **The structured encoder approach:** In this approach, structure is imposed on the encoder which generates the code words.

For obtaining general block codes to be used for BCM, a new approach, called **the structured distance approach**, is proposed in the thesis. This approach is based on the observation of the following facts:

- Coded modulation over AWGN channels uses the Euclidean distance metric as against the Hamming distance metric used in the conventional codes.

- For a general channel signal constellation the Hamming distance and the Euclidean distance are not equivalent.

- In conventional codes, since the Hamming distance metric is used, the code words (mapped to labels) have the information of the distance. This is not so for the case of Euclidean distances. In general, for an arbitrary signal constellation it might not be possible to map the signals with labels such that the code words have the information of the Euclidean distances.

- For a channel signal constellation, generally the number of elements in the set of Euclidean distances between the signals is much less than the number of signals in the channel signal constellation.

- The encoder generates the code words, but the coding problem is to search for optimum codes for certain applications. The main aim is to maximize the minimum Euclidean distance. For a general problem using an arbitrary signal constellation it cannot be assured that a particular system will result in the maximization of the minimum Euclidean distance between the code words.

These observations provide the primary motivation for a change of viewpoint to the coding problem. The new point of view results in **the structured distance approach**. Salient features of the structured distance approach for code search are as follows:

(1) No structure is assumed on the code words.

(2) No specific type of encoder is assumed before obtaining the codes.

(3) No specific structure is assumed for the channel signal constellation.

(4) Euclidean distances between the signals provided by an expanded channel signal constellation are used.

(5) The set of Euclidean distances provided by an expanded channel signal constellation and the d_{min} required for the BCM scheme results in an Euclidean distance distribution for the code.

(6) Code search consists of obtaining code words such that the minimum Euclidean distance between the code words is $\geq d_{\text{min}}$.

(7) Euclidean distances between the channel signals are the prime entities and no consideration is given to the soft-decoding complexity of the code, for the code search.

The structured distance approach to the coding problem shifts the perspective from the code words or sequences of signals of a channel signal constellation to the Euclidean distance distribution or sequences of Euclidean distances between signals provided by the channel

signal constellation. Irrespective of what the code words turn out to be, coding has to just assure that all the elements in the distance distribution are $\geq d_{\min}$.

General (non-linear) block codes which can be used with BCM schemes and which need not be linear, cyclic, group, lattice, geometrically uniform [GU] or rectangular codes, are obtained using the structured distance approach. A wide range of codes with various rates and having redundancy in space only, or in space and time, are reported based on this new approach. Unlike various existing techniques, the structured distance approach to the coding problem has a wider applicability as it can be used with any type of expanded channel signal constellation. General codes based on asymmetric expanded channel signal constellations and channel signal constellations with number of signals not equal to a prime or a power of prime are found.

A new class of general block codes obtained by the structured distance approach in conjunction with some well known results from sphere packings is given. The problem of finding codes for BCM, as in the case of error-correcting codes, is analogous to the sphere packing problem [21]. The problem of obtaining codes for a BCM scheme over AWGN channels can be stated to be "finding a finite number of code words (points) in a finite Euclidean discrete space such that the minimum Euclidean distance $d_{\min}$ between the code words (points) is maximized." In order to obtain a sphere packing corresponding to the coding problem specified by the application, though a packing which increases the contact number might not be the most dense packing in general, for the specific case of the bounded discrete Euclidean space of a BCM scheme, attention is restricted to packings which increase the contact number in a finite volume. This requires an arrangement which maximizes the sum of contact numbers of the spheres. Such an arrangement is used as the condition for structuring the Euclidean distances to obtain the distance distribution of a code, resulting in a new class of codes belonging to the general codes of the structured distance approach. A rotationally symmetric arrangement of spheres which increases the contact number with a central sphere is obtained by considering all the permutations of a valid distance n-tuple. The distance distribution of a code is obtained by selecting permutations of valid distances. The class of general codes obtained in this manner is called **codes based on selective permutations of distances.**

Once a code suitable for a specific application is found using the structured distance approach, the block encoder can be designed for transforming data words into code words. Two schemes are given in the thesis for the design and implementation of block encoders for general (non-linear) block codes.

(1) Block encoder using the code table.

(2) Block encoder using an intermediate binary logic stage.

Maximum-likelihood soft-decoding of block codes may be performed by Viterbi algorithm which uses the trellis of the block code. In such a scheme, since the required data memory storage is directly proportional to the number of vertices in the trellis, minimization of the number of vertices in the trellis at every stage of the algorithm is of primary concern. This provides one of the basic motivations for the study and use of minimal trellises of block codes [61]. Equivalent codes sometimes may yield a more efficient trellis, called the **optimal minimal trellis**, than the minimal trellis of the original code. For a general block code, the optimal minimal trellis might not be unique, but a set of optimal minimal trellises can exist. Further, the minimal trellis or the optimal minimal trellis need not be proper. Code trees of equivalent codes are used to obtain optimal minimal proper trellises for general (non-linear) block codes. The problem of obtaining optimal minimal improper trellises for general block codes, of block length 3, is also considered. The schemes given in the thesis for obtaining optimal minimal trellises are for general (non-linear) codes which might not be linear, cyclic, lattice, group, GU or rectangular codes.

A new scheme for soft decoding of general block codes used for BCM is proposed. It uses a reduced tree in place of trellis and is shown to offer various trade-offs. The resulting decoder is amenable to parallel implementation and is shown to be more suitable for a block code. Further, back tracking necessary in a soft decoder based on the Viterbi algorithm, is avoided by using the code tree instead of the trellis.

For soft decoding, in general, it is necessary to compute Euclidean distances of a received word with the code words and store these distances for comparison. In the tree representation of a code, the Euclidean distances are computed using the edges of the tree along various paths and are known as path metrics. The path metrics along various paths are stored at the vertices of the tree.

The main disadvantage of the tree representation is that the number of vertices, and hence the storage required for data, grows rapidly with the size of the code. To reduce this growth and to make the tree suitable for representation of block codes, it is necessary to reduce the code tree obtained for a BCM scheme. Reduction of the code tree for soft decoding of codes for BCM reduces the storage requirements. To reduce the storage requirements for the code tree, it is necessary to eliminate as many vertices from consideration at lower levels in the code tree as possible. Elimination of a vertex of a tree from further consideration implies that the subtree rooted at that vertex is eliminated. When a vertex is eliminated from further consideration, no computations are performed for the subtree rooted at that vertex. The reduction of the storage is, of course, achieved at the cost of reduction in the parallelism of the tree. In this way a trade-off exists between parallelism and memory storage for a tree based soft decoder. Depending on the specific need of the application a proper configuration of the soft decoder can be designed.

The following schemes have been given for reduction of code trees:

(1) Reduction based on subtrees in the code tree isomorphic with weights.

(2) Reduction based on using the property of code equivalence.

(3) Reduction based on the values of the distance previously computed at level n in the tree during soft decoding.

Concatenation of BCM schemes is used to obtain general block codes of long length using general block codes of short lengths. Both the inner and the outer codes correspond to BCM schemes and use general (non-linear) codes with Euclidean distance metric. The inner block code is considered as a virtual expanded channel signal constellation by the outer block code. Block codes of length 3 are considered for concatenation. In general for a q-stage concatenated scheme, $q > 2$, a block code of length 3^q, will be obtained. In such a scheme, the number of code words and hence the number of data words depends on the selection of the individual BCM schemes, whether redundancy is added in space only or in space and time both. The code search is simplified, since instead of searching for a block code of length $n = 3^q$, $q > 2$, it is necessary to search for codes of length 3 only, though for a larger virtual channel signal constellation. Concatenation also simplifies the implementation of the block encoder, as encoding can proceed in stages in a concatenated BCM-BCM scheme.

Each stage has to deal with 3-tuples of signals from the signal constellation provided by the previous stage. Finally, the soft decoder at each decoding stage has to handle decoding of block codes of length 3 only, thereby significantly reducing the decoding complexity. In this way concatenation provides a trade-off between time (length of the block code n) and space (the number of signals in the virtual channel signal constellation).

The work done provides various options and trade-offs for efficient design and implementation of BCM systems. It may be of special interest when coded modulation is to be utilized for some application using block codes of short block length.

Contents

List of Figures

List of Tables

List of Notations

n	block length of a block code or length of a sequence.
S	set of Euclidean distances between signals.
n'	number of signals in a signal constellation.
S'	set of signals forming the expanded channel signal constellation.
s_i'	signal belonging to a signal constellation.
$d(s_i', s_j')$	Euclidean distance between signals s_i' and s_j'.
$d_{i,j}$	Euclidean distance between signals s_i' and s_j'.
N'	number of distinct distances between various signals of a signal constellation.
S	set of signals of a constellation and the Euclidean distances between them.
$\mathrm{d_S}$	matrix representation for the Euclidean distances between signals of a signal constellation.
$S' \times S'$	set of two-tuples of signals from a signal constellation.
$s_i's_j'$	two-tuples of signals from a signal constellation.
$d(s_i's_j', s_k's_l')$	Euclidean distance between two-tuples of signals $s_i's_j'$ and $s_k's_l'$.
$d_{ij,kl}$	same as $d(s_i's_j', s_k's_l')$.
(Ed)	Euclidean distance operator.
$\emptyset$	empty set.

n'_b	number of signals in the base signal constellation.
B'	base signal constellation.
n'	number of signals in the expanded channel signal constellation.
d_{min}	minimum Euclidean distance between code words.
d_{uc}	Euclidean distance for uncoded modulation.
C	set of block codes for a BCM scheme.
$\|C\|$	number of code words for a BCM scheme.
G	asymptotic coding gain.
R	rate of a block code.
D	set of all Euclidean distances between signals of a channel signal constellation.
$\circ$	distance composition operator.
d_k	same as $d_{i,j}$, Euclidean distance between two signals from the signal constellation.
$P(D)$	power set of the set D.
$\hat{D}_{S'\times S'\times\cdots\times S'(\text{n-times})}$	set of valid Euclidean distances for the distance distribution of a code of block length n.
$\tilde{n}$	dimensionality of the Euclidean space of the code words.
$\tilde{n}_b$	dimensionality of the Euclidean space of the uncoded data words.
N_b	number of points in the discrete finite Euclidean space of the data words.
N_c	number of points in the discrete finite Euclidean space of the code words.
$\tilde{c}$	set of relevant spheres.
τ_i	contact number of sphere $\tilde{c}_i$ with center c_i.
V	volume of the $\tilde{n}$ dimensional sphere of radius $d_{\text{min}}/2$.
V_{s_i}	volume of the i^{th} sphere lying inside the $\tilde{n}$-dimensional cube.
V_i	volume of the i^{th} Voronoi region inside the $\tilde{n}$-dimensional cube.
$\tilde{S}_{d_{i_1},d_{i_2},\cdots,d_{i_n}}$	set of all permutations of $d_{i_1}, d_{i_2}, \cdots, d_{i_n}$.
$v_{i,j}$	j^{th} vertex at level i of the code tree.

$\mathbf{T}$	code tree.
$\mathbf{T}_{v_{i,j}}$	subtree rooted at vertex $v_{i,j}$.
$d_{v_{i,j}}$	path metric (Euclidean distance) stored at vertex $v_{i,j}$ of the code tree.
l_i	number of vertices at level i in the code tree.
$\mathbf{V}_{i,k}$	k^{th} partition set of the vertices at level i of the code tree.
$\overset{\text{w}}{\cong}$	relation denoting that two subtrees are isomorphic with weights.
$\mathbf{RV}_{i,k}$	set of vertices obtained after reducing the tree by eliminating vertices from $\mathbf{V}_{i,k}$.
$m_{i,k}$	cardinality of set $\mathbf{V}_{i,k}$.
$\mathbf{D}_{i,k}$	set of the path metrics at the vertices in $\mathbf{V}_{i,k}$.
$\mathbf{P}_{i,k}$	poset $\{\mathbf{D}_{i,k} \leq\}$.
$\mathbf{MP}_{i,k}$	set of minimal elements of $\mathbf{P}_{i,k}$.
$lP_{i,k}$	minimal element of $\mathbf{P}_{i,k}$.
α	permutation of the n co-ordinate positions of a code word to obtain an equivalent code.
$IV_{i,j}$	edge of the tree in to vertex $v_{i,j}$ of the tree.
$OV_{i,j,q}$	q^{th} edge of the tree out of vertex $v_{i,j}$ of the tree.
$\mathbf{IV}_{i,k,p}$	p^{th} edge of the trellis in to the vertex $\mathbf{V}_{i,k}$ of the trellis.
$\mathbf{OV}_{i,k,q}$	q^{th} edge of the trellis out of the vertex $\mathbf{V}_{i,k}$ of the trellis.
$\tilde{\mathbf{T}}$	set of minimal proper trellises for all the equivalent codes of a block code used with a BCM scheme.
C_i	for all values of i is the set of the equivalent codes.
$\bar{S}_{i,0}$	set of symbols at co-ordinate position 0 in the code word.
$\bar{S}_{i,2}$	set of symbols at co-ordinate position 2 in the code word.
o_i	i^{th} block in a concatenated BCM scheme.
$\|V\|$	number of signals in the virtual channel signal constellation.

Chapter 1

Introduction

In this age of information more and more people desire to access the latest information on various topics as comfortably as possible. Exchanging information has become an essential part of life. Systems dealing with the exchange of information are known as **Communications Systems**. Various systems exist for a reliable exchange of audio and visual information.

Since the advent of computer systems, and their becoming popular, it became necessary for the computers also to exchange information. As computers are digital devices, **Digital Communication** became important and systems were developed for exchanging information which is termed as data.

The medium through which this exchange of information takes place is known as channel, it is intermediate between the source and the sink of information. Some examples of channels in common use are air, space, telephone cables, optical cables and television cables. The channel in addition to carrying information has associated with it certain unwanted signals known as noise. It is necessary for the signal generated at the source to undergo various transformations. **Information Theory** deals with this modification of information for combating noise and making it suitable for transmission.

The challenges of this field are the development of techniques, for achieving faster and reliable communication, using existing technology.

1.1 Genesis of Coded Modulation

This section briefly presents the historical background and describes the evolution of digital communication systems. A survey of relevant literature is also interspersed.

1.1.1 Basic Digital Communication System

A basic digital communication system, shown in Figure 1.1.1, consists of a source of digital signals known as transmitter, which sends the information and a sink of digital signals known as the receiver. For efficient transmission the signal has to be modified into a form which is suitable for being sent through the channel. This is achieved by using a modulator and a

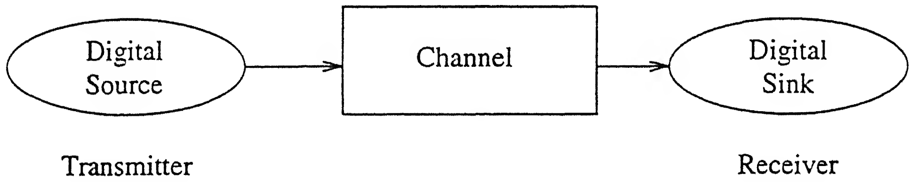

Figure 1.1.1: Basic digital communication system

demodulator as shown in Figure 1.1.2. The output of the modulator which is transmitted over the channel is known as the channel signal. The demodulator recovers back digital signals. Modulation is performed using various techniques such as amplitude shift keying [ASK], phase shift keying [PSK] and frequency shift keying [FSK], in which the amplitude, phase and frequency of a sine wave carrier signal are changed according to the digital bit being transmitted. For data transmission at higher rates, multileveled channel signals are used, some of which are given in Appendix A.

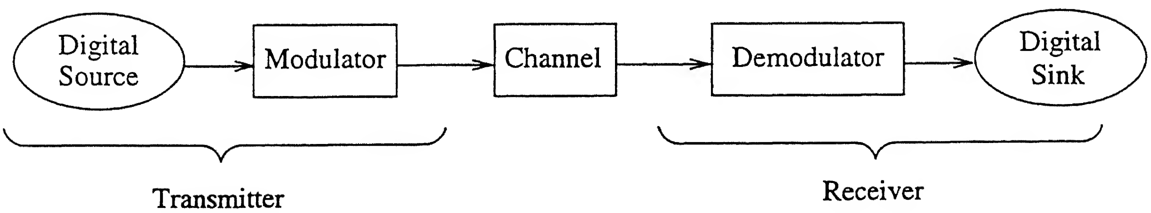

Figure 1.1.2: Digital communication system with modulation

Channels are attributed by various parameters. A few types of channels relevant with the thesis are described here:

Power-limited channels: The transmitted signal power cannot be increased arbitrarily.

Band-limited channels: The rate of transmission of signals over these channels is limited.

Discrete/continuous, source/sink channels: These are channels depending on the nature of source and the sink connected to the channel.

Fading channels: The signal over these channels undergoes fading e.g. wireless communication.

Additive white Gaussian noise [AWGN] channels: Channels on which the noise is additive and its nature is white Gaussian.

Memory-less channels: These are channels which do not have memory.

Erasure channels: Channels in which burst of disturbances can erase digits of information.

For binary signals transmitted over the channel the bit error rate [BER] and for non-binary signals the symbol error rate [SER] characterize the errors occurring in transmission due to noise. This is also specified in terms of probability of error Pr(e). Another important term is the strength of the signal being transmitted over a noisy channel as compared to the noise on the channel, specified by the signal to noise ratio [SNR] of the channel. Most of the terminology followed in the thesis is as discussed in the books by Haykin [37,38], Proakis [64] and Lathi [54]. Viterbi and Omura [86], and, Wozencraft and Jacobs [91] are also classical references on these topics.

In digital communication over noisy channels, when errors occur, two strategies are commonly employed:

(1) Detect errors at the receiver and if errors occur request the transmitter to retransmit the message.

(2) Detect and correct errors at the receiver.

The second strategy, known as forward error correction [FEC], is of prime concern in this thesis.

To incorporate FEC in digital communication systems, a channel encoder and channel decoder have to be included as shown in Figure 1.1.3. The pioneering work of **C. E.**

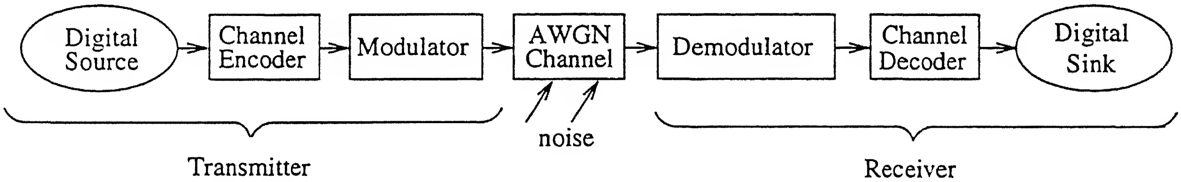

Figure 1.1.3: Digital communication system with error control

Shannon [72, pages 379–423 and 623–656] and [73, 74] propelled this interesting branch of communication and information theory in which a rich stream of research continues to provide technological achievements, encompassing new applications offering wider and better communication. Shannon's channel coding theorem states that with sufficient but finite redundancy, properly introduced in the data by a channel encoder, it is possible for a channel decoder to reconstruct the input data sequence to any degree of accuracy desired. If the rate of data input to a channel encoder is less than a given value, known as the channel capacity, then the encoding and decoding operations can result in an error-free reconstruction of the input data sequence. As a consequence of this, coding theory developed channel encoders and decoders, to improve the reliability of transmission by suitably adding redundancy to channel symbols.

Traditionally coding and modulation are performed as separate operations and the introduction of redundancy by the channel encoder increases the transmission bandwidth as more bits have to be transmitted. Various techniques related to this can be found in text books by Viterbi and Omura [86], MacWilliams and Sloane [58], Blahut [16], and, Lin and Costello [56].

Source encoding and source decoding are also used at the transmitter and receiver, respectively for efficient representation of the source output so that the redundancy in the data is reduced.

1.1.2 Introduction to Coded Modulation

Consider the case of band-limited, power-limited channels in which both the bandwidth and the signal power are constrained. In such channels what is the efficient means of communication in the presence of additive white Gaussian noise?

Conventional coding techniques add redundant bits to the data to combat noise. This results in more number of bits being transmitted on the channel, requiring more bandwidth. In the absence of additional bandwidth, the alternative is to use a larger channel signal constellation. The extra bits are accommodated in the larger symbol space and as a result the symbol transmission does not require more bandwidth. But, this results in a larger probability of error [Pr(e)] at the same SNR because now the separation (distance) between the signals is reduced. To achieve the same Pr(e), the signal power has to be increased. So for a band-limited and power-limited channel, how can efficient transmission be achieved?

The answer to this problem is **Coded Modulation**. Massey [59] and Ungerboeck [78] showed that for band-limited, power-limited channels combining coding and modulation at the transmitter and combining decoding and demodulation at the receiver can be used to obtain better performance. Coded modulation deals with combining these blocks in the digital communication system as shown in Figure 1.1.4. The encoder is designed using the

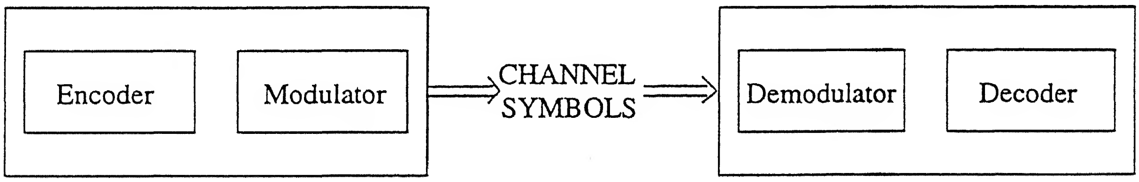

Figure 1.1.4: Coded modulation system

channel symbols which are the output of the modulator, and the decoder instead of using quantized bits, directly uses the symbols received from the channel. Coded modulation suggests that for digital transmission over analog channels which are band-limited and power-limited, it is more efficient to have in the transmitter and receiver the channel encoder and decoder which use analog signals as output and input respectively. Due to this the encoders and decoders can be viewed as mappers between digital signals and analog signals, and vice versa.

A comprehensive summary of the techniques relating to certain issues on coded modulation are discussed in the book by Biglieri et al. [15]. The August 1989 and December 1989 issues of *"IEEE Journal of selected areas in communications"* are devoted to coded modulation. Besides various survey articles on the topic can be found in the *"IEEE Communications magazine"*, for example, Ungerboeck [79, 80] and Viterbi et al. [85].

1.1.3 Characteristics of Coded Modulation

Coded modulation techniques which are used for band-limited power-limited channels can be characterized by the following properties:

(1) Expanded channel signal constellation.

(2) Redundancy in signal space.

(3) Soft decoding.

Expanded Channel Signal Constellation:
In coded modulation the encoding is performed with channel signals which are analog. Instead of considering analog signal space, Ungerboeck [78] showed that, approximately the same Pr(e) can be achieved by doubling the size of the channel signal constellation. The modulator outputs twice the number of channel signals as required for the uncoded case and it is not necessary to consider analog channel signals. This is helpful in the search for codes to be used to design the encoder. But the objective is to have analog input and analog output channel and still keep the complexity low enough, as is indicated with the advent of coded modulation systems with larger and larger channel signal constellations.

Redundancy in Signal Space:
In traditional coding, redundancy is added in time as more bits are added to the data word to get the code word. Coded modulation provides another dimension for adding redundancy, i.e., the signal space. Larger number of channel signals and the spatial separation between them in the signal space are used to provide redundancy. Coded modulation systems can be classified in two types:

(1) Systems (power-limited, band-limited channels) with redundancy in signal space only.

(2) Systems with redundancy in both time and signal space.

Coded modulation provides trade off of how much redundancy can be put in time and space. This issue is dealt in greater detail later on in the thesis.

Soft Decoding:
In coded modulation redundancy is added in space and sequences of redundant space are used to obtain proper distances. Coding and decoding takes place using sequences known as data word and code word. For the purpose of decoding unquantized channel symbols have to be considered over a sequence of particular length. As no hard decisions are made at the demodulator, this is known as soft decoding. Since decoding involves finding a transmitted sequence closest to the received sequence, it is also known as maximum likelihood [ML] decoding.

Any coded modulation system is characterized by these properties of combining coding and modulation and soft decoding using redundancy in space over sequences of signals.

With coded modulation even source coding can be combined, so that source coding, channel coding and modulation are all done together as discussed by Ayanoğlu et al. [3] and Vaishampayan et al. [81].

1.1.4 Coded Modulation Systems

In this section a classifications of coded modulation systems is given. These cover a wide range of applications.

1.1.4.1 Classification Based on Channel Characteristics

Systems for AWGN memory-less channels:
In AWGN channels, when a finite number of waveforms are used to transmit information, the measure of the similarity or dissimilarity of the set of signal waveforms is the Euclidean distance between the waveforms [64]. When a sequence of such waveforms corresponding to sequence of symbols is used, the Euclidean distances between these sequences denote the separation or the distance. When the channel signal constellations are such that the Euclidean distance and Hamming distance are equivalent e.g. 2-PSK, then the codes defined with the Hamming distance metric can be used, but in general for AWGN channels the Euclidean

distance metric has to be considered. The classical coding schemes like Hamming codes, BCH codes, Reed-Muller codes, Hadamard codes, Golay codes and Nordstrom-Robinson codes, etc. are but a particular case of codes with the Hamming distance metric. In coded modulation the coding problem is searching code words such that the minimum Euclidean distance between the code words, which are sequences of signals, is maximized.

The decoding of the codes has to be soft maximum likelihood. In this, a sequence of signals is received and its Euclidean distance with all the code words is found. The code word which is at the minimum Euclidean distance is declared as the decoded code word.

Telephone lines and television cables are examples of AWGN memory-less channels.

When the carrier phase of the signal at the receiver is perfectly known the detection is known as coherent and when the phase is ignored at the receiver, we have non-coherent reception. When coherent detection is required with suppressed carrier modulation, sometimes to compensate for a phase ambiguity in the receiver, codes have to be designed which also have a property that they are transparent to phase offsets at multiples of the smallest differences between two modulation angles in the signal constellation. Such codes are known as rotationally invariant codes. Wei [89] has obtained rotationally invariant trellis coded M-PSK multidimensional schemes. The problem of rotational invariance of codes is not further dealt in this thesis.

Systems for fading channels:

A typical example of the fading channel is the mobile satellite communication channel. In this the bandwidth is constrained as many users in a cell have to share the bandwidth and the power is limited due to the power constraint on the radiation system and the physical size of the mobile antenna. Such channels have Doppler frequency shifts due to vehicular motion, multi-path fading and shadowing. In the case of most of the mobile satellite channels, fading is modeled by a Rician amplitude probability distribution. In heavy terrain where severe shadowing is a problem, the channel is characterized by Rayleigh fading. For such systems coded modulation can be used for efficient signaling. A summary of coded modulation techniques and their application to faded channels have been discussed by Jamali [45].

1.1.4.2 Classification Based on Modulator Characteristics

The modulator provides signals which are transmitted over the channel. For achieving high bit rates, the trend is towards using large channel signal constellations, for example, the V.34 standard uses a 240 point signal constellation. In the literature there are references of various modulation techniques like amplitude shift keying [ASK] [15, 78], phase shift keying [PSK] [5, 9, 10, 28, 42, 65, 78], quadrature amplitude modulation [QAM] [32, 69, 78] and various asymmetric modulation techniques, [15].

Multi-dimensional modulations:

The basic objective is to accommodate more and more signal points and have large distances between them. As the number of dimensions grows, there is more space to accommodate the signals and hence the distance between the signal points increases. So multidimensional modulation techniques are more efficient for coded modulation. Multidimensional constellations have a smaller constituent two-dimensional constellation, easier tolerance to phase ambiguities and a better trade-off between complexity and coding gain. Though this is at the cost of a more complex modulator and demodulator.

In [32], it is shown that multidimensional constellations are desirable for representing fractional number of bits per two dimensions, useful for increasing SNR efficiency, and natural for use with multidimensional coded modulation. Desirable characteristics of such a constellation include good SNR efficiency, low implementation complexity, compatibility with coded modulation and with QAM modems, including small size and peak-to-average power ratio of its constituent 2-d constellation, phase symmetry, scalability, and capability of supporting a secondary channel. In [27], Voronoi constellations which are implementable N dimensional constellations, based on partitions of N dimensional lattices that can achieve good shape gains and that are inherently suited for use with coded modulation are discussed.

1.1.4.3 Classification Based on Encoder Characteristics

Convolutional encoder:

If the encoder used for coded modulation is a convolutional encoder, then the scheme is known as **trellis coded modulation [TCM]**. This is the scheme which has been popularly used for coded modulation since the eighties and still continues to be used in even the recent modems over telephone lines. This scheme has the name TCM, since a trellis is used

for the soft decoding of the codes. Various standards for modems including the CCITT's V.34 use TCM in their modem design for error correction. The advantage of this scheme is the simplicity of its practical implementation in both encoding and decoding. The basic disadvantage is that a long decoding delay has to be tolerated and good codes generally tend to be long. Also if the AWGN channel is bursty, then convolutional codes are not sufficient to combat burst errors.

The classic paper by Ungerboeck [78], generated the spurt of research in TCM. This paper gives coded modulation schemes for efficient transmission over power-limited band-limited channels using binary convolutional codes, with a mapping of coded bits into channel signals termed as set-partitioning. Wei [88], presented schemes which start with a variety of multidimensional lattices and result in trellis codes. Calderbank and Sloane [20], provide alternative method to Ungerboeck's set partitioning for constructing trellis codes using signal constellations consisting of points from a n-dimensional lattice. Zehavi and Wolf [92] have used generating function techniques for analyzing the error event and the bit error probabilities for trellis codes. Benedetto et al. [11], give theoretical aspects of combined coding and modulation and results on linear codes and a subset of linear codes known as super-linear codes. Codes for 16- and 32-PSK are found and their performance evaluation has been carried out. Forney [28], introduced a more general class of group codes that have symmetry properties, called as geometrically uniform [GU] codes. These codes are more general than Slepian-type group codes [76] or lattice codes [25, 26].

A group code C over a group G is a set of sequences of group elements that itself forms a group under a component wise group operation. A study of the state spaces, trellis diagrams and canonical encoders is done by Forney [29]. Benedetto et al. [9] have applied the theory of GU trellis codes to multidimensional PSK constellations. Symmetry group of $L \times$ MPSK constellation is completely characterized and conditions for rotational invariance of geometrically uniform partitions of a signal constellation are given. Geometrically uniform partitions are obtained which are used as starting points in a search for good trellis codes based on binary convolutional codes. Using these, good geometrically uniform trellis codes over non-binary abelian groups, are constructed in [10]. As no simple mathematical formulas exist to determine the best code for a given application, extensive computer runs are used to find and evaluate the most suitable codes. Benedetto et al. [12] show how to evaluate TCM schemes, describing the parameters and the algorithms used. Baldini and Farrell [7],

give a coded modulation scheme based on multilevel convolutional codes defined over finite rings of integers. Rossin et al. [69], introduce trellis group codes as an extension of Slepian group codes [76], to codes over sequence spaces. A class of rotationally invariant, non-abelian trellis group codes have been developed for QAM modulation schemes.

Block encoder:

If the encoder used for coded modulation is a block encoder then, the scheme is called **block coded modulation** [BCM]. Codes of short lengths and finite decoding delay can be obtained using this scheme. Nonlinear block codes can also be used in coded modulation schemes. Their main disadvantage is that if we need a block code of very long length, then the code search and soft decoding complexity makes them unsuitable as compared to convolutional codes. This thesis mainly deals with nonlinear BCM.

Imai and Hirakawa [41], proposed a scheme for multilevel coding using several error-correcting codes. The codes used have different error-correcting capabilities and efficient systems can be obtained by choosing these codes appropriately. The codes are soft decoded. The paper by Forney [31], presents a comprehensive tutorial survey of the development of efficient modulation techniques for band limited channels, such as telephone channels, and reviews TCM and BCM. Sayegh [70], has given a class of optimum signal space codes based on the Euclidean distance metric constructed using short binary block codes. Forney [25, 26], characterizes a large number of coded modulation techniques (including lattice codes and TCM), proposed for band limited channels. A unified development of the family of lattices that have proved to be most useful in constructing coset codes is discussed. Properties of such lattices like, their minimum squared distances, their partitions and aspects of their structure that are useful in decoding are given.

Muder [61] describes minimal proper trellises for block codes. It is shown that the minimal proper trellises exist for all block codes. Bounds on the trellis size of linear block codes are given by Berger and Be'er [14]. Kschischang et al. [50] give schemes to design block codes for M-ary PSK where M has the form $2^k \times 3^l$. Conditions under which known block codes designed for discrete symmetric channels (i.e. for the Hamming metric) may be mapped onto M-PSK schemes for the AWGN channel are investigated. Buda [17], demonstrate that the optimal channel codes need not be random and some of them have structure, e.g., the structure of a lattice code. The Leech lattice was suggested for practical implementation

of BCM modems by Lang and Longstaff [53]. Be'ery [8] show that BCM has performance comparable to TCM. An efficient algorithm for maximum likelihood soft decoding of Leech lattice is obtained.

Kasami et al. [46], use multilevel techniques for combining block coding and modulation. Given a multilevel block modulation code with no interdependency among the binary component codes, the proposed method gives a multilevel block modulation code that has the same rate and a minimum squared Euclidean distance not less than that of the original code and a smaller number of nearest neighbor code words. Loeliger [57], addresses the general problem of matching signal sets to generalized linear algebraic codes. Burr [18], in an introductory article, give the principles of coded modulation, describing and comparing BCM and TCM. Multilevel coding methods based on rings of integer modulo-q (q is a non-prime integer number), which are suitable for coded modulation using q-PSK and q-QAM, are obtained by Baldini and Farrell [6]; for systematic circulant linear block, pseudo-cyclic multilevel block, phase-invariant multilevel circulant block and phase-invariant multilevel quasi-circulant block codes. In [5] a class of multilevel block codes based on fields are presented, based on block sub codes of adequate length. The scheme is shown to have advantages over coded modulation based on rings of integers.

Isaksson and Zetterberg [42], give block-coded M-PSK modulation, in which the expanded signal-set is given the structure of a finite field. The code is defined by a square nonsingular circulant generator matrix over the field. The codes are described using trellises and soft decoded using the Viterbi algorithm. Caire and Biglieri [19] give trellis construction, bounds on the minimum Euclidean distance and some examples of coded modulation schemes based on prime power order cyclic groups. In analogy with the usual construction of linear codes over fields, a generator matrix and a parity check matrix is also defined for these codes. Rifà [67] describe block codes to be used with two-dimensional QAM signal constellations. Garello and Benedetto [33] discuss the problem of building multilevel group codes, i.e., codes obtained combining separate coding at different levels, in such a way that, the resulting code is a group code. Construction leading to multi-level group codes for semi-direct and direct products is illustrated, and geometrically uniform Euclidean-space codes obtained from multilevel codes over abelian and non abelian groups are given.

Lafourcade and Vardy [52], give lower bounds on trellis complexity of block codes. The problem of minimizing the vertex count at a given time index in the trellis for a general (nonlinear) code is shown to be NP-complete by Kschischang and Sorokine [51]. It is shown that the minimal trellis for a nonlinear code may not be observable i.e., some code words may be represented by more than one path through the trellis, and that minimizing the vertex count at one time index may be incompatible with minimizing the vertex count at another time index. Piret [63], obtain linear cyclic codes with good Euclidean distances using 8-PSK signal constellation considered as a ring of integers. Vazirani, Saran and Sundar Rajan [83]; Vardy and Kschischang [82]; and Kschischang [49] have contributed recently on obtaining minimal trellises for block codes.

Based on the work by Kasami et al. [47, 48] on the permutations of the bit positions, recently Horn and Kschischang [40] and Schuurman [71] have reported interesting facts for linear codes in obtaining efficient trellis representation.

Concatenated encoder:

The encoder can also be formed by concatenating two or more encoders. Various possible combinations are described.

Convolutional-Convolutional encoder: This consists of a concatenation of two convolutional encoders. Such systems can be used for obtaining very long length coded modulation schemes. No references are available on the practical utilization of such systems.

Convolutional-block encoder: The encoder consists of a concatenation of a block encoder with a convolutional encoder. The block code can be viewed as an virtual signal space. This scheme can be used where we need very large signal constellations and long codes. Rajpal et al. [65, 66] have used binary convolutional codes with good free branch distances as outer codes and block MPSK modulation codes as inner codes.

Block-convolutional encoder: This type of concatenation is useful, when a inner convolutional code provides for a efficient use of bandwidth and an outer block code or BCM scheme can be used to take care of burst noise. If the convolutional code used is long, then the decoding delay, and the complexity of the block code, increases. Pellizzoni and Spalvieri [62] discuss the encoding and decoding for binary multilevel

coset codes obtained by concatenating convolutional outer codes and Reed-Muller inner codes. Deng and Costello [22] give concatenated scheme with TCM as inner code and Reed-Solomon codes as outer codes. Also, as discussed by Benedetto and Montorsi [13], a class of codes which use blocks of convolutional encoders concatenated in parallel with interleavers to obtain long length codes known as **turbo codes** are of interest.

Block-block encoder: If a block encoder is concatenated with another block encoder then we can have codes of long lengths with simple decoding and low encoding complexity. Also the BCM-BCM scheme can select various block codes to deal with various channels impairment like for e.g. a bursty telephone line or a bursty fading channel. Herzberg et al. [39] discuss a concatenated scheme with lattice codes as inner codes and Reed-Solomon codes with hard-decision decoding as outer codes. Such encoders are dealt with later on in this thesis.

1.1.4.4 Classification Based on Decoder Characteristics

At the receiving end of the channel the demodulator, which receives the channel signals, is followed by a decoder. The decoder is used to obtain the data words from the code words after error correction. Broadly, the decoders can be classified as follows.

Hard decoder: In a hard decoder the demodulator quantizes the received channel symbol. The channel information is lost. The demodulator makes decisions and the decoder can make use of the algebraic decoding schemes utilizing the structure of the code words.

Soft decoder: The demodulator does not quantizes the received channel symbol. The channel information and noise is preserved. The demodulator does not makes any decisions. It is the decoder which makes use of all the received information and makes the decision regarding the received word.

1.1.5 Soft Decoding

Whatever type of coded modulation system which may be used, the receiver is almost invariably a maximum likelihood decoder. This avoids hard quantizations and provides with channel information which can be used by the decoder to make decisions over a sequence of received symbols. For decoding at the receiver, the encoding scheme and the code table information are required. The mapping between the data words and the code words for coded modulation schemes is generally represented using a trellis diagram. The received word is matched with the code words and a most likely estimate is found, the data word corresponding to that code word is the received decoded word. For a sequential soft decoding of symbols, stage by stage, using a trellis, the Viterbi algorithm [24.84] is used.

Bahl et al. [4], give an optimal decoding strategy for linear block and convolutional codes for minimizing the symbol error rate. Wolf [90], describes schemes for soft decision maximum likelihood decoding of linear block codes using the Viterbi algorithm. It is shown that for cyclic codes the trellis is periodic. Matis and Modestino [60] present an algorithm for soft decoding of linear block codes by a reduced complexity search through a trellis derived from the parity check matrix.

The complexity of the decoder and the memory storage required for the decoder are also important criteria in the selection of a proper encoding scheme. At times good encoding schemes cannot be used as their soft decoding can be impractical.

The ideas of coded modulation have been applied to the area of vector quantization. Work has been reported in the literature on trellis coded vector quantization by Fischer [23] and Wang [87]. This has not been discussed further in this thesis.

1.1.6 Modem

Modems are indispensable for digital communication over telephone lines. Over the years, governed by the need, and the progress of the technology, various standards have been developed for modems. CCITT's recent standard V.34 and the architectural aspects of the state of the art modems have been discussed here.

A modem is a unit, consisting of a modulator and a demodulator, which are required to transmit and receive data over a telephone line. The strongly market-driven modem business,

in the last three decades, has resulted in a steady development of faster and better modems. For the proper development of modems by various manufacturers, CCITT (International Telephone and Telegraph Consultative Committee) has come up with various standards over the years [34]. These are listed in Table 1.1.1.

Table 1.1.1: Details of some CCITT standards

Year	Standard	Data rate b/s	Coding	Modulation
1964	V.21	300	–	FSK
1972	V.27	4800	Scrambler	8-phase DPSK
1976	V.29	up to 9600	Scrambler	16-pt. QAM
1984	V.32	up to 9600	Scrambler	32-pt. TCM
1991	V.32bis	up to 14400	up to 128-pt. TCM	
1994	V.fast	up to 24000	precoding with multi-point 4-d TCM	
1995	V.34	up to 28800	4-d 16, 32 or 64 state Wei code	

A Brief Note on the V.34 Standard:

This new standard is defined with a maximum signaling rate of 28.8 kbits/s, operating over the general switched telephone network, and over two-wired leased lines. Modems conforming to this standard are complex in design due to the fast data rates and the large amount of flexibility requirements. V.34 prescribes slower speeds in the steps of 2400 bits/s, for example 26.4 kbits/s, 24 kbits/s, etc., if required due to adverse line conditions.

V.34 implements a complex set-up scheme in which almost all of the modem parameters are configured at the connection time. It uses QAM with trellis coded modulation. V.34 specifies the coding as 4-dimensional 16-state Wei code, 4-dimensional 32-state Wei code, or 4-dimensional 64-state Wei code, with fractional bit rates. These offer differences in terms of immunity to Gaussian noise traded off against the complexity of the decoder. For its implementation V.34 requires approximately 35 to 40 MIPS of signal processing power.

Implementation of Modems:

It is up to the manufacturer to make architectural decisions on how to implement each function and which functions to group within a particular chip. These decisions directly affect the performance, flexibility, and cost of the chip or the chip set being produced.

The actual modem function is usually referred to as a data pump, a device that translates digital information into the properly modulated waveforms for transmission and performs the inverse function on the incoming data. Some data pumps employ a specialized array of dedicated analog and digital logic circuits to perform this task, while others use general purpose digital signal processing [DSP] chips. If a DSP-based data pump is chosen, a coder-decoder [CODEC] function is required to translate between the digital format of the processor and the analog line signal. A separate line-interface function performs signal conditioning, isolation, and impedance matching between the data pump and the telephone line.

A controller looks after the overall operation of the modem, provides directions to the data pump and controls the flow of data to and from the host-computer system. The controller usually performs error control, data compression, communication with the host-computer system, and line-control functions. In addition to this, the controller is usually responsible for implementing a command set that permits the host-computer system to control the modem. Most often, the Hayes modem "AT" command set is used as a de facto standard.

The host interface is serial for external application and is parallel when the modem is mounted on the motherboard or on the bus. Nowadays PCMCIA-interface is also provided. The recent developments of the V.34 modem standard permits full-duplex transmission at rates extended up to 33.6 kbits/s over the public switched telephone network. Forney et al. [30] briefly describe the technologies that are used to make these bit rates possible. This new high-speed enables various multimedia modem applications.

Besides the standard modems sometimes modems have to be designed for specific applications, in which certain specific design considerations have to be satisfied. Also there is a wide network of television cables, which are being proposed to be used for digital communication and providing inter-networking capability. For this, cable modems, which work at a much faster rates for practical implementations, are required.

1.2 Motivation

The previous sections gave a brief introduction and discussed various coded modulation schemes in use. From this overview, it is observed that coded modulation originated in the seventies and was primarily meant for efficient signaling over band-limited, power-limited

channels. The use of coded modulation system has its origin in trellis coded modulation [TCM], and presently a large market of modems for telephone lines for achieving high bit rates, continues to use TCM. Although, block coded modulation [BCM] evolved simultaneously, it is still in an experimental stage and does not finds wide usage[1] in general modems. Recently, there has been renewed interest in BCM and trellises for block codes following the work done by various authors, for example, Isaksson et al. [42], Herzberg et al. [39], Baldini et al. [6], Lafourcade et al. [52], Kschischang et al. [51], Vardy et al. [82], Vazirani et al. [83] and Kschischang [49]. This section gives the motivation behind the work presented in this thesis and how it evolved from the related investigations.

The objective of the work is to provide a generalized framework for BCM and consider schemes for code search, encoding and decoding. Such an investigation is important for the following reasons.

- Most of the work reported on BCM is based on some particular signal constellations. The schemes proposed are valid for PSK [5,9,10,28,42,65,78], QAM [32,69,78], or some other specific signal constellations. The coding should, in fact, be applicable for any modulation technique which can be used. A generalized framework is required which provides a technique, which can be applied to a wide range of signal constellations.

- Schemes for linear BCM [42,63] have been proposed, which with long length codes give performance comparable to TCM. As such a linear block code of long length can be considered to be equivalent to a convolutional code and hence there does not seem to be much reason for probing this path. So this thesis considers schemes for nonlinear codes. In fact, the codes considered here are more general than linear codes [86], cyclic codes [58], lattice codes [28], group codes [29], geometrically uniform [GU] codes [28] or rectangular codes [49], reported in literature.

- TCM is widely used in modems over telephone lines. The major disadvantage with TCM is that, in the decoding the sequence has to be terminated after a long length and hence the delay involved is long. Compared to this, BCM can provide schemes which use block codes of short lengths and of a fixed delay.

[1]Some applications like the Leech lattice modem [53], are exceptions

- The major disadvantage of BCM is the complexity and the memory required by the encoding and decoding schemes. But with the massive developments on the technological front this barrier of complexity is fast crumbling and various schemes which can have the advantage of using parallel processing provided by block codes can be obtained. So BCM can be used for fast media like cable modems where decoding delay has to be minimized and kept fixed.

The existing block coding techniques, use the Hamming metric and are not suitable for most applications of coded modulation over AWGN channels. Though recently there has been renewed interest in considering the trellises of linear and nonlinear block codes and using these codes with BCM, these codes do not assume a band-limited. power-limited channel, but they are based on a binary symmetric channel assumption. In these codes redundancy is in time, while for coded modulation it is in signal space. No generalized schemes have been proposed till now for applications in which a trade-off between redundancy in time and signal space can be utilized to find codes which are in between these two extremes.

Concatenated block coding schemes exist and are generally used over bursty channels. These schemes can also be used to obtain block codes of long length in a systematic manner. This basically reduces the encoding and decoding complexity. Most of the schemes proposed in the literature [22, 39], use binary codes with a coded modulation code for concatenated schemes.

The above discussion can be summarized with the following observations which provide the necessary direction as well as motivation for the present work. The work in coded modulation has predominantly evolved from TCM using convolutional codes and schemes for BCM are being considered. Due to this the applications for BCM and also a viewpoint exists which is biased by convolutional codes. Also the traditional block coding schemes, a wide variety of which exists, are being used to fit into a convolutional code like framework under the name of BCM. This thesis considers a different perspective of treating BCM as a separate entity. The problem is considered in a general framework which provides a rich variety of trade-offs in various situations which can be utilized in an efficient manner.

The work presents a generalized framework for block coded modulation [BCM]. A set-theoretic approach is employed to arrive at this generalization. Certain encoding and decoding schemes based on this framework are also given. As an outcome, some trade-offs become

important in designing, encoding and decoding BCM systems for various applications.

1.3 Thesis Organization

This thesis is concerned with the developing of a generalized theoretical framework for the code search, encoding and soft decoding of block coded modulation [BCM] schemes over AWGN memory-less channels. This section briefly describes the organization and major contributions of the thesis.

Chapter 2, discusses signal spaces, which are spaces of signals from a constellation with the Euclidean distance metric. The concept of virtual signal constellations is introduced, which are, in fact, BCM code words. General set-theoretic representations for the signal constellation and the Euclidean distances between signals are given and properties of some signal constellations are studied. A representation scheme for all finite sequences of signals from a signal constellation is discussed. A representation of the Euclidean distances between all the sequences, which is termed as the Euclidean distance distribution of sequences over finite signal constellations, is obtained which is compact. An arbitrary channel signal constellation is defined which is used in the schemes of this thesis.

Chapter 3 uses the representation of signals and Euclidean distances between them, obtained in Chapter 2, for the purpose of coding. A structured distance approach for BCM is proposed as opposed to the existing structured code approach and the structured encoder approach. This is based on the use of sequences of Euclidean distances for obtaining the Euclidean distance distribution and then the code words. A distance distribution description of code words is obtained from which the encoder is designed. This change in viewpoint imparts the approach with the generality which is desired. The codes obtained are general (non-linear) as the prime objective is of ensuring the satisfaction of a distance distribution and the scheme can be used with an arbitrary channel signal constellation. Algorithms and examples illustrating the scheme are given.

Chapter 4 uses the structured distance approach proposed in Chapter 3 with certain important results from sphere packings to obtain a class of the general block codes, using the structured distance approach, known as codes based on the selective permutations of distances. The scheme described here can also be used with an arbitrary channel signal

constellation. Certain properties for codes used with BCM are obtained by considering their analogy with sphere packings. A condition is obtained and is used in conjunction with relevant assumptions for symmetric arrangements of spheres, to obtain a scheme which results in the class of codes known as codes based on the selective permutation of distances. This scheme results in general (non-linear) codes over arbitrary channel signal constellations. Algorithms and illustrative examples are discussed.

Chapter 5 considers the problem of soft decoding of the nonlinear block codes obtained by the schemes discussed in Chapters 3 and 4. The set of code words is represented using a code tree. Various techniques for reducing the code tree such that the implementation of the soft decoder becomes practical are discussed. The advantage of using the reduced tree for soft decoding is that a parallel implementation of the decoder is possible. Considerations regarding the memory storage and the comparisons required for the soft decoder are also further explored. Algorithm and examples of the scheme are given.

Chapter 6 discusses the problems associated with obtaining efficient trellis representations for the soft decoding of general (non-linear) codes. Code equivalence is used and schemes for obtaining the optimal minimal proper trellis for any general code, and the optimal minimal improper trellis for any general code of block length 3 are discussed. The Viterbi algorithm is used with the trellis for soft decoding of general (non-linear) block codes. Algorithm and examples for the schemes proposed for obtaining the trellises are give.

Chapter 7 discusses concatenated coding schemes for BCM. Here, concatenation is primarily used to obtain general (non-linear) block codes of long length in a systematic manner from block codes of short length 3. The major advantage is that the code search, encoder and the soft decoder are simplified. BCM is used as both inner and outer codes and the Euclidean distance metric is employed for coding. The stages after the first stage use a virtual channel signal constellation. This is possible due to the generality of the schemes proposed in the previous chapters. The chapter concludes with algorithms and illustrative examples demonstrating this scheme.

Chapter 8 concludes the thesis, where the work is subjected to a critical appraisal, highlighting the major contributions and mentioning the limitations of the approach. Some ideas for future work are also presented.

The thesis also includes three appendices. **Appendix A** gives the details regarding the various channel signal constellations considered in this thesis. **Appendix B** consists of a listing of various general codes, obtained by the schemes discussed in the thesis, for a variety of expanded channel signal constellations and for concatenated schemes. **Appendix C** gives a brief overview of some results from sphere packings.

Chapter 2

Set-theoretic Representation for Signal Constellations and Euclidean Distances

2.1 Introduction

Coded modulation uses an expanded channel signal constellation to perform coding. This chapter presents a frame work for representation of arbitrary channel signal constellations which include virtual channel signal constellations. Important properties of some example channel signal constellations are studied. A representation for Euclidean distances between signal sequences is found.

A communication channel with discrete inputs, transmits a finite set of signals which convey information. For a power-limited channel, the channel signal energy is limited. For a band-limited channel. the rate at which the signals can be transmitted is limited. Depending on these constraints a proper channel signal constellation is selected. The modulator uses the channel signal constellation and selects a particular channel signal from it to be transmitted at a particular instant.

In coded modulation an expanded channel signal constellation is used. For AWGN channels, coding is performed using these channel signals of the signal constellation with the Euclidean distance metric. Ungerboeck [78] stated that by doubling the number of channel signals, almost all is gained in terms of channel capacity that is achievable by signal-set

expansion if at a given SNR satisfactory error performance can no longer be achieved by uncoded modulation. At times, considering other issues, having channel signal constellations which are not just double in size can lead to various simplifications. For the code search encoding and decoding, it is advantageous to minimize the number of signals if it does not compromises with other parameters of interest.

Code words are sequences of channel signals of some finite length n, selected from the set of channel signal constellation, where n is known as the block length of the code. The basic coding problem for an AWGN channel is finding sufficient code words such that the minimum Euclidean distance between them is maximized and is larger than the distance for the uncoded case. A brute force approach to coding will be to find the Euclidean distances between all the sequences of signals of a channel signal constellation of a particular length, and properly select the appropriate sequences as codewords. This chapter is devoted to the study of the Euclidean distances between signals and signal sequences for various types of signal constellations. It creates the basis for a general framework which is developed in Chapter 3. The issues regarding the tables of Euclidean distances between signals and sequences of signals are relevant in the efficient implementation of various algorithms proposed in the later chapters.

2.2 Euclidean Distances Between Signals of a Signal Constellation

Consider a finite set of n' signals forming a signal constellation[1],

$$S' = \{\, s'_0,\ s'_1,\ s'_2,\ \ldots,\ s'_{(n'-1)} \,\}.$$

Definition 2.2.1 *The distance between any two signals in S' is defined as,*

$$d(s'_i, s'_j) = d_{i,j} = \sqrt{(s'_i - s'_j)^2} \quad 0 \le i, j \le (n'-1).$$

Theorem 2.2.1 *The set S' with the distance d forms a metric space.*

[1]Since a general frame work is to be developed, signal constellations dealt with are constellations which practically might not correspond to a modulator scheme, these are termed as signal constellations and not channel signal constellations.

Proof: The distance d, satisfies the following properties for every pair of elements of S', (s'_i, s'_j):

(1) $d_{i,j} = d_{j,i}$,

(2) $d_{i,j} > 0$ if $i \neq j$ $d_{i,j} = 0$ if $i = j$,

(3) $d_{i,j} \leq d_{i,k} + d_{k,j}$.

Hence, d defined over the set S', satisfies the axioms of a metric.
Hence ,the set S' with the distance d forms a metric space. □

The distance d, defined in the Definition 2.2.1, is known as the **Euclidean distance metric** between signals.
Let N' denote the total number of distinct distances between various elements of the set S'. Then,

$$N' = (n'-1) + (n'-2) + \ldots + 1 = \frac{(n'-1)n'}{2}.$$

From the elements of the set S' and the Euclidean distances between them, another set S is created, which is as follows,

$$S \text{ s.t. } s_{i,j} \in S \quad \forall\, 0 \leq i,j \leq (n'-1) \text{ and } s_{i,j} = (s'_i, s'_j, d_{i,j})$$

This set S, is the set of signals of a signal constellation and the Euclidean distances between them. This is a set theoretic representation of the signal constellation.
This set can be represented as a table of Euclidean distances between signals.

d	s'_0	s'_1	$\ldots$	$s'_{n'-1}$
s'_0	$d_{0,0}$	$d_{0,1}$	$\ldots$	$d_{0,n'-1}$
s'_1	$d_{1,0}$	$d_{1,1}$	$\ldots$	$d_{1,n'-1}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$s'_{n'-1}$	$d_{n'-1,0}$	$d_{n'-1,1}$	$\ldots$	$d_{n'-1,n'-1}$

Since the Euclidean distance between signals is a metric,

$$\forall i\; d_{i,i} = 0 \text{ and } \forall i,j\; d_{i,j} = d_{j,i}.$$

Hence, in the table there are only N' different distances and all the diagonal elements are zero. The matrix $\mathbf{d_S}$ obtained from the table is a more suitable form of representation for

using with algorithms discussed later on in the thesis. The Euclidean distances between signals of a signal constellation in a matrix form is,

$$\mathbf{d_S} = \begin{bmatrix} d_{0,0} & \cdots & d_{0,n'-1} \\ \vdots & \ddots & \vdots \\ d_{n'-1,0} & \cdots & d_{n'-1,n'-1} \end{bmatrix}. \tag{2.2.1}$$

The matrix represented by the table is symmetric. Also, if the signal set has some structural symmetries, these will be reflected on the structure of the matrix. Due to this, reductions can be achieved in the storage of Euclidean distances for signal constellations.

Example 2.2.1 Consider the 8-PSK signal constellation given in Appendix A.

The $n' = 8$ signals of the channel signal constellation are denoted as:
$s'_0 = 0$, $s'_1 = 1$, $s'_2 = 2$, $s'_3 = 3$, $s'_4 = 4$, $s'_5 = 5$, $s'_6 = 6$, $s'_7 = 7$.
The set of signals is,

$$S' = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ \}.$$

Calculating the Euclidean distances between the signals of the 8-PSK channel signal constellation, the set S is as follows,
$S = \{\ (0,0,0),\ (0,1,\sqrt{0.59}),\ (0,2,\sqrt{2}),\ (0,3,\sqrt{3.41}),\ (0,4,2),\ (0,5,\sqrt{3.41}),$
$(0,6,\sqrt{2}),\ (0,7,\sqrt{0.59}),\ (1,0,\sqrt{0.59}),\ (1,1,0),\ (1,2,\sqrt{0.59}),\ (1,3,\sqrt{2}),$
$(1,4,\sqrt{3.41}),\ (1,5,2),\ (1,6,\sqrt{3.41}),\ (1,7,\sqrt{2}),\ (2,0,\sqrt{2}),\ (2,1,\sqrt{0.59}),$
$(2,2,0),\ (2,3,\sqrt{0.59}),\ (2,4,\sqrt{2}),\ (2,5,\sqrt{3.41}),\ (2,6,2),\ (2,7,\sqrt{3.41}),$
$(3,0,\sqrt{3.41}),\ (3,1,\sqrt{2}),\ (3,2,\sqrt{0.59}),\ (3,3,0),\ (3,4,\sqrt{0.59}),\ (3,5,\sqrt{2}),$
$(3,6,\sqrt{3.41}),\ (3,7,2),\ (4,0,2),\ (4,1,\sqrt{3.41}),\ (4,2,\sqrt{2}),\ (4,3,\sqrt{0.59}),$
$(4,4,0),\ (4,5,\sqrt{0.59}),\ (4,6,\sqrt{2}),\ (4,7,\sqrt{3.41}),\ (5,0,\sqrt{3.41}),\ (5,1,2),$
$(5,2,\sqrt{3.41}),\ (5,3,\sqrt{2}),\ (5,4,\sqrt{0.59}),\ (5,5,0),\ (5,6,\sqrt{0.59}),\ (5,7,\sqrt{2}),$
$(6,0,\sqrt{2}),\ (6,1,\sqrt{3.41}),\ (6,2,2),\ (6,3,\sqrt{3.41}),\ (6,4,\sqrt{2}),\ (6,5,\sqrt{0.59}),$
$(6,6,0),\ (6,7,\sqrt{0.59}),\ (7,0,\sqrt{0.59}),\ (7,1,\sqrt{2}),\ (7,2,\sqrt{3.41}),\ (7,3,2),$
$(7,4,\sqrt{3.41}),\ (7,5,\sqrt{2}),\ (7,6,\sqrt{0.59}),\ (7,7,0),\ \}.$
The table of Euclidean distances between signals obtained from the set S is as shown in Table 2.2.1.

Table 2.2.1: Euclidean distances between signals of 8-PSK constellation for Example 2.2.1

d	0	1	2	3	4	5	6	7
0	0	$\sqrt{0.59}$	$\sqrt{2.00}$	$\sqrt{3.41}$	2	$\sqrt{3.41}$	$\sqrt{2.00}$	$\sqrt{0.59}$
1	$\sqrt{0.59}$	0	$\sqrt{0.59}$	$\sqrt{2.00}$	$\sqrt{3.41}$	2	$\sqrt{3.41}$	$\sqrt{2.00}$
2	$\sqrt{2.00}$	$\sqrt{0.59}$	0	$\sqrt{0.59}$	$\sqrt{2.00}$	$\sqrt{3.41}$	2	$\sqrt{3.41}$
3	$\sqrt{3.41}$	$\sqrt{2.00}$	$\sqrt{0.59}$	0	$\sqrt{0.59}$	$\sqrt{2.00}$	$\sqrt{3.41}$	2
4	2	$\sqrt{3.41}$	$\sqrt{2.00}$	$\sqrt{0.59}$	0	$\sqrt{0.59}$	$\sqrt{2.00}$	$\sqrt{3.41}$
5	$\sqrt{3.41}$	2	$\sqrt{3.41}$	$\sqrt{2.00}$	$\sqrt{0.59}$	0	$\sqrt{0.59}$	$\sqrt{2.00}$
6	$\sqrt{2.00}$	$\sqrt{3.41}$	2	$\sqrt{3.41}$	$\sqrt{2.00}$	$\sqrt{0.59}$	0	$\sqrt{0.59}$
7	$\sqrt{0.59}$	$\sqrt{2.00}$	$\sqrt{3.41}$	2	$\sqrt{3.41}$	$\sqrt{2.00}$	$\sqrt{0.59}$	0

The matrix of Euclidean distances is as follows,

$$\mathbf{d_S} = \begin{bmatrix} 0 & \sqrt{0.59} & \sqrt{2.00} & \sqrt{3.41} & 2 & \sqrt{3.41} & \sqrt{2.00} & \sqrt{0.59} \\ \sqrt{0.59} & 0 & \sqrt{0.59} & \sqrt{2.00} & \sqrt{3.41} & 2 & \sqrt{3.41} & \sqrt{2.00} \\ \sqrt{2.00} & \sqrt{0.59} & 0 & \sqrt{0.59} & \sqrt{2.00} & \sqrt{3.41} & 2 & \sqrt{3.41} \\ \sqrt{3.41} & \sqrt{2.00} & \sqrt{0.59} & 0 & \sqrt{0.59} & \sqrt{2.00} & \sqrt{3.41} & 2 \\ 2 & \sqrt{3.41} & \sqrt{2.00} & \sqrt{0.59} & 0 & \sqrt{0.59} & \sqrt{2.00} & \sqrt{3.41} \\ \sqrt{3.41} & 2 & \sqrt{3.41} & \sqrt{2.00} & \sqrt{0.59} & 0 & \sqrt{0.59} & \sqrt{2.00} \\ \sqrt{2.00} & \sqrt{3.41} & 2 & \sqrt{3.41} & \sqrt{2.00} & \sqrt{0.59} & 0 & \sqrt{0.59} \\ \sqrt{0.59} & \sqrt{2.00} & \sqrt{3.41} & 2 & \sqrt{3.41} & \sqrt{2.00} & \sqrt{0.59} & 0 \end{bmatrix}.$$

The matrix $\mathbf{d_S}$ is circulant due to the symmetries in the signal constellation. To store this matrix it is sufficient to store only one row of Euclidean distances from which any required distance can be computed. Appendix A gives the matrices for the square of the Euclidean distances between signals of various channel signal constellations used in the examples of this thesis. Due to the power constraint, all the points corresponding to the signals are assumed to lie within the unit circle. Table 2.2.2 summarizes various characteristic attributes of these signal constellations. For various channel signal constellations the number of signals and the set of Euclidean distances are tabulated in this table. The column **Min. Ed.** gives the minimum Euclidean distance between the nearest neighboring signals belonging to the constellation and **No. ne.** tabulates the number of nearest neighbors. The nature of the matrix $\mathbf{d_S}$, depending on the symmetries inherent in the signal constellation are given in the column **Nature of matrix** and the column **% Redu. in store** gives the reduction in

Table 2.2.2: Euclidean distances between signals for various signal constellations

Sr. No.	Constellation	No. of signals	Set of Euclid. distances	Min. Ed.	No. ne.	Nature of matrix	% Redu. in store
1	2-PSK	2	$\{2\}$	2	1	Circulant	75
2	3-PSK	3	$\{\sqrt{3}\}$	$\sqrt{3}$	2	Circulant	88.89
3	4-PSK	4	$\{\sqrt{2}, 2\}$	$\sqrt{2}$	2	Circulant	81.25
4	Asy 4-PSK	4	$\{1, \sqrt{3}, 2\}$	1	1	Symm. present	75
5	5-PSK	5	$\{\sqrt{1.38}, \sqrt{3.62}\}$	$\sqrt{1.38}$	2	Circulant	84
6	6-PSK	6	$\{1, \sqrt{3}, 2\}$	1	2	Circulant	86.1
7	7-PSK	7	$\{\sqrt{0.75}, \sqrt{2.45}, \sqrt{3.8}\}$	$\sqrt{0.75}$	2	Circulant	87.76
8	8-PSK	8	$\{\sqrt{0.59}, \sqrt{3.41}, 2\}$	$\sqrt{0.59}$	2	Circulant	89.06
9	Asy 8-PSK	8	$\{\sqrt{0.27}, \sqrt{2}, \sqrt{3}, 2, 1$ $\sqrt{3.73}\}$	$\sqrt{0.27}$	1	Various symm. present	75
10	8-AM PM	8	$\{\sqrt{0.88}, \sqrt{0.44}, \sqrt{2.2},$ $\sqrt{1.77}, 2, 3\}$	$\sqrt{0.44}$	4	Various symm. present	75
11	16-PSK	16	$\{\sqrt{0.15}, \sqrt{0.59}, \sqrt{1.23},$ $\sqrt{2}, \sqrt{2.76}, \sqrt{3.41},$ $\sqrt{3.85}, 2\}$	$\sqrt{0.15}$	2	Circulant	94.14
12	16-QAM	16	$\{\sqrt{0.22}, \sqrt{0.88}, \sqrt{1.98},$ $\sqrt{2.2}, \sqrt{1.1}, \sqrt{0.44},$ $\sqrt{1.77}, \sqrt{2.86}, 2\}$	$\sqrt{0.22}$	4	Various symm. present	75

the storage for the matrix of the square of Euclidean distances due to the symmetries of the matrix.

2.3 Euclidean Distances between Signal Sequences

Code words of block length n are sequences of signals from a signal constellation, of finite length n, with the constraint that the minimum Euclidean distance between them is maximized. So it is of interest to study the Euclidean distances between signal sequences.

Consider all the signal sequences of length two, formed from the signals of a signal constellation. The set of the two-tuples of signals is denoted as,

$$S' \times S' = \{\, s'_0 s'_0,\ s'_0 s'_1,\ \ldots,\ s'_{n'-1} s'_{n'-1} \,\}.$$

Definition 2.3.1 *A distance between any two two-tuples is defined as,*

$$d(\, s'_i s'_j,\ s'_k s'_l \,) = d_{ij,kl} = \sqrt{(s'_i - s'_k)^2 + (s'_j - s'_l)^2} \quad \forall 0 \le i, j, k, l \le (n' - 1).$$

Theorem 2.3.1 *The set $S' \times S'$ with the distance d forms a metric space.*

Proof: The distance d, satisfies the following properties for every pair of two-tuples, $(s'_i s'_j, s'_k s'_l)$:

(1) $d_{ij,kl} = d_{kl,ij}$,

(2) $d_{ij,kl} > 0$ if $ij \neq kl$ $d_{ij,kl} = 0$ if $ij = kl$,

(3) $d_{ij,kl} \le d_{ij,mn} + d_{mn,kl}$.

Hence, d defined over the set $S' \times S'$ satisfies the axioms of a metric.
Hence, the set $S' \times S'$ with the distance d forms a metric space. □

The distance d, defined in the Definition 2.3.1 is known as the **Euclidean distance metric** between signals sequences of length two.

Lemma 2.3.1 $d_{ij,kl} = d_{ji,lk} = d_{kj,il} = d_{jk,li} = d_{il,kj} = d_{li,jk} = d_{kl,ij} = d_{lk,ji}$.

Proof: The result follows trivially from the Definition 2.3.1. □

Definition 2.3.2 *The operation of Definition 2.3.1 can also be denoted as,*

$$d_{ij,kl} \triangleq d_{i,k}(Ed)d_{j,l}. \tag{2.3.1}$$

Note that, (Ed) is a binary operator such that,

$$\begin{aligned} d_{i,k}(Ed)d_{j,l} &= \sqrt{d_{i,k}^2 + d_{j,l}^2} \\ &= \sqrt{(s_i' - s_k')^2 + (s_j' - s_l')^2} \quad \forall 0 \leq i, j, k, l \leq (n' - 1). \end{aligned}$$

The distances between the sequences $s_k's_i'$ and $s_k's_j'$, $\forall\ i, j$ s.t. $0 \leq i, j \leq (n' - 1)$, are represented as a matrix $d_{kk}(Ed)\mathbf{d_S}$. This matrix is of the same order as the order of matrix $\mathbf{d_S}$. The distances between the sequences $s_i's_j'$ and $s_k's_l'$, $\forall\ i, j, k, l$ s.t. $0 \leq i, j, k, l \leq (n' - 1)$, are represented as a matrix $\mathbf{d_S}(Ed)\mathbf{d_S}$. The order of this matrix is the square of the order of the matrix $\mathbf{d_S}$.

This representation is structurally similar to the Kronecker product of matrices, the operation involved here is different and it is not matrix product.

Example 2.3.1 Consider the 2-PSK signal constellation given in Appendix A.

The $n' = 2$ signals of the channel signal constellations are s_0' and s_1'.
The set of signals is, $S' = \{\, s_0',\ s_1' \,\}$.
Calculating the Euclidean distances between the signals of the 8-PSK channel signal constellation, the set S is as follows,

$$S = \{(s_0', s_0', 0), (s_0', s_1', 2), (s_1', s_0', 2), (s_1', s_1', 0)\}.$$

The Euclidean distances between signals obtained from the set S are shown in Table 2.3.1. The matrix of Euclidean distances is as follows,

$$\mathbf{d_S} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}.$$

Table 2.3.1: Euclidean distances between signals of 2-PSK constellation for Example 2.3.1

d	s'_0	s'_1
s'_0	0	2
s'_1	2	0

Consider all the signal sequences of length two, formed from the channel signals of a signal constellation.
The set of the two-tuples of signals is,

$$S' \times S' = \{\, s'_0 s'_0,\ s'_0 s'_1,\ s'_1 s'_0,\ s'_1 s'_1,\, \}.$$

The Euclidean distance between signal sequences, $s'_0 s'_0$ and $s'_0 s'_1$ is,

$$d(s'_0 s'_0, s'_0 s'_1) = d_{00,01} = d_{0,0}(Ed)d_{0,1} = 2.$$

The Euclidean distances between the sequences $\{\, s'_0 s'_0, s'_0 s'_0 \,\}$, $\{\, s'_0 s'_0, s'_0 s'_1 \,\}$, $\{\, s'_0 s'_1, s'_0 s'_0 \,\}$ and $\{\, s'_0 s'_1, s'_0 s'_1 \,\}$ can be represented in the matrix form as $d_{s'_0 s'_0}(Ed)\mathbf{d}_\mathbf{S}$.
This is,

$$0(Ed)\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}.$$

The Euclidean distances between all the sequences,
$\{\, s'_0 s'_0, s'_0 s'_0 \,\}$, $\{\, s'_0 s'_0, s'_0 s'_1 \,\}$, $\{\, s'_0 s'_0, s'_1 s'_0 \,\}$, $\{\, s'_0 s'_0, s'_1 s'_1 \,\}$, $\{\, s'_0 s'_1, s'_0 s'_0 \,\}$, $\{\, s'_0 s'_1, s'_0 s'_1 \,\}$, $\{\, s'_0 s'_1, s'_1 s'_0 \,\}$, $\{\, s'_0 s'_1, s'_1 s'_1 \,\}$, $\{\, s'_1 s'_0, s'_0 s'_0 \,\}$, $\{\, s'_1 s'_0, s'_0 s'_1 \,\}$, $\{\, s'_1 s'_0, s'_1 s'_0 \,\}$, $\{\, s'_1 s'_0, s'_1 s'_1 \,\}$, $\{\, s'_1 s'_1, s'_0 s'_0 \,\}$, $\{\, s'_1 s'_1, s'_0 s'_1 \,\}$, $\{\, s'_1 s'_1, s'_1 s'_0 \,\}$ and $\{\, s'_1 s'_1, s'_1 s'_1 \,\}$ can be represented in the matrix form as, $\mathbf{d}_\mathbf{S}(Ed)\mathbf{d}_\mathbf{S}$. This is,

$$\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}(Ed)\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 4 & 2 \\ 2 & 4 & 0 & 2 \\ 4 & 2 & 2 & 0 \end{bmatrix}.$$

Theorem 2.3.2 *For a sequence of signals of length n from the finite set S' forming a channel signal constellation, the Euclidean distances between all the n-tuples can be represented by the matrix,*

$$\mathbf{d}_{\mathrm{S}\times\mathrm{S}\times\ldots\times\mathrm{S}\,(n\text{-}times)} = \mathbf{d}_{\mathrm{S}}(Ed)\mathbf{d}_{\mathrm{S}}\ldots(Ed)\mathbf{d}_{\mathrm{S}\,(n\text{-}times)}.$$

Proof: Consider signal sequences of length 2, based on Equation 2.3.1,

d	$s'_0\ s'_0$	$s'_0\ s'_1$	$\cdots$	$s'_{n'-1}\ s'_{n'-1}$
$s'_0\ s'_0$	$d_{00,00}$	$d_{00,01}$	$\cdots$	$d_{0(n'-1),0(n'-1)}$
$s'_0\ s'_1$	$d_{00,01}$	$d_{00,11}$	$\cdots$	$d_{0(n'-1),1(n'-1)}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$s'_{n'-1}\ s'_{n'-1}$	$d_{(n'-1)0,(n'-1)0}$	$d_{(n'-1)0,(n'-1)1}$	$\cdots$	$d_{(n'-1)(n'-1),(n'-1)(n'-1)}$

Using the matrix representation of Equation 2.2.1 and from Definition 2.3.1, this can be written as,

$$\begin{aligned}\mathbf{d}_{\mathrm{S}\times\mathrm{S}} &= \begin{bmatrix} d_{0,0}(Ed)\mathbf{d}_{\mathrm{S}} & \cdots & d_{0,n'-1}(Ed)\mathbf{d}_{\mathrm{S}} \\ \vdots & \ddots & \vdots \\ d_{n'-1,0}(Ed)\mathbf{d}_{\mathrm{S}} & \cdots & d_{n'-1,n'-1}(Ed)\mathbf{d}_{\mathrm{S}} \end{bmatrix} \\ &= \mathbf{d}_{\mathrm{S}}(Ed)\mathbf{d}_{\mathrm{S}}\end{aligned}$$

If for some $n = k$, this representation is valid then,

$$\mathbf{d}_{\mathrm{S}\times\mathrm{S}\times\ldots\times\mathrm{S}(\text{k-times})} = \mathbf{d}_{\mathrm{S}}(Ed)\mathbf{d}_{\mathrm{S}}\ldots(Ed)\mathbf{d}_{\mathrm{S}(\text{k-times})}. \tag{2.3.2}$$

Now for $n = k+1$,

$$\begin{aligned}\mathbf{d}_{\mathrm{S}\times\ldots\times\mathrm{S}(\text{k+1-times})} &= \begin{bmatrix} d_{0,0}(Ed)\mathbf{d}_{\mathrm{S}\times\ldots\times\mathrm{S}(\text{k-times})} & \cdots & d_{0,n'-1}(Ed)\mathbf{d}_{\mathrm{S}\times\ldots\times\mathrm{S}(\text{k-times})} \\ \vdots & \ddots & \vdots \\ d_{n'-1,0}(Ed)\mathbf{d}_{\mathrm{S}\times\ldots\times\mathrm{S}(\text{k-times})} & \cdots & d_{n'-1,n'-1}(Ed)\mathbf{d}_{\mathrm{S}\times\ldots\times\mathrm{S}(\text{k-times})} \end{bmatrix} \\ &= \mathbf{d}_{\mathrm{S}}(Ed)\mathbf{d}_{\mathrm{S}\times\mathrm{S}\times\ldots\times\mathrm{S}(\text{k-times})}.\end{aligned}$$

From Equation 2.3.2, this can be written as,

$$\mathbf{d}_{\mathrm{S}\times\mathrm{S}\times\ldots\times\mathrm{S}(\text{k+1-times})} = \mathbf{d}_{\mathrm{S}}(Ed)\mathbf{d}_{\mathrm{S}}\ldots(Ed)\mathbf{d}_{\mathrm{S}(\text{k+1-times})}.$$

Hence by mathematical induction, in general for sequences of signals of a finite length n we have,

$$\mathbf{d}_{\mathrm{S}\times\mathrm{S}\times\ldots\times\mathrm{S}(\text{n-times})} = \mathbf{d}_{\mathrm{S}}(Ed)\mathbf{d}_{\mathrm{S}}\ldots(Ed)\mathbf{d}_{\mathrm{S}(\text{n-times})}.$$

□

The matrix $\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\ldots\times\mathbf{S}(\text{n-times})}$ gives the Euclidean distance distribution for the sequences of signals of length n from a signal constellation. Each element of the matrix corresponds to particular distances of the distance distribution. This theorem gives a compact representation for the Euclidean distance distribution. From this representation the number of signal sequences at a particular distance from a given signal sequence of length n can also be calculated.

Example 2.3.2 Consider a signal constellation consisting of four signals represented by the following set,

$$S' = \{\, s'_0,\ s'_1,\ s'_2,\ s'_3 \,\}.$$

The Euclidean distance between the signals of the signal constellation are specified as follows,
$S = \{\, (s'_0, s'_0, 0),\ (s'_0, s'_1, \sqrt{2}),\ (s'_0, s'_2, 2),\ (s'_0, s'_3, \sqrt{2}),\ (s'_1, s'_0, \sqrt{2}),\ (s'_1, s'_1, 0),$
$(s'_1, s'_2, \sqrt{2}),\ (s'_1, s'_3, 2),\ (s'_2, s'_0, 2),\ (s'_2, s'_1, \sqrt{2}),\ (s'_2, s'_2, 0),\ (s'_2, s'_3, \sqrt{2}),$
$(s'_3, s'_0, \sqrt{2}),\ (s'_3, s'_1, 2),\ (s'_3, s'_2, \sqrt{2}),\ (s'_3, s'_3, 0) \,\}.$
We have the following matrix of the Euclidean distance between the signals,

$$\mathbf{d}_{\mathbf{S}} = \begin{bmatrix} 0 & \sqrt{2} & 2 & \sqrt{2} \\ \sqrt{2} & 0 & \sqrt{2} & 2 \\ 2 & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix}.$$

The matrix $\mathbf{d}_{\mathbf{S}}$ is circulant due to the symmetries in the signal constellation which is, in fact, the QPSK signal constellation.
Table 2.3.2 shows the Euclidean distances between signal sequences of length 2. In the table for the sake of simplicity of representation,
$s'_0 = 0,\ s'_1 = 1,\ s'_2 = 2$ and $s'_3 = 3$.

$$\begin{aligned} \mathbf{d}_{\mathbf{S}\times\mathbf{S}} &= \begin{bmatrix} 0(Ed)\mathbf{d}_{\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} & 2(Ed)\mathbf{d}_{\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} \\ \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} & 0(Ed)\mathbf{d}_{\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} & 2(Ed)\mathbf{d}_{\mathbf{S}} \\ 2(Ed)\mathbf{d}_{\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} & 0(Ed)\mathbf{d}_{\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} \\ \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} & 2(Ed)\mathbf{d}_{\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}} & 0(Ed)\mathbf{d}_{\mathbf{S}} \end{bmatrix} \\ &= \begin{bmatrix} 0 & \sqrt{2} & 2 & \sqrt{2} \\ \sqrt{2} & 0 & \sqrt{2} & 2 \\ 2 & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix} (Ed) \begin{bmatrix} 0 & \sqrt{2} & 2 & \sqrt{2} \\ \sqrt{2} & 0 & \sqrt{2} & 2 \\ 2 & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix}. \end{aligned}$$

Table 2.3.2: Euclidean distances between signal sequences of length 2 for Example 2.3.2

d	00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33
00	0	$\sqrt{2}$	2	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{6}$	2	2	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{2}$	2	$\sqrt{6}$	2
01	$\sqrt{2}$	0	$\sqrt{2}$	2	2	$\sqrt{2}$	2	$\sqrt{6}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{8}$	2	$\sqrt{2}$	2	$\sqrt{6}$
02	2	$\sqrt{2}$	0	$\sqrt{2}$	$\sqrt{6}$	2	$\sqrt{2}$	2	$\sqrt{8}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{6}$	2	$\sqrt{2}$	2
03	$\sqrt{2}$	2	$\sqrt{2}$	0	2	$\sqrt{6}$	2	$\sqrt{2}$	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	2	2	$\sqrt{6}$	2	$\sqrt{2}$
10	$\sqrt{2}$	2	$\sqrt{6}$	2	0	$\sqrt{2}$	2	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{6}$	2	2	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$
11	2	$\sqrt{2}$	2	$\sqrt{6}$	$\sqrt{2}$	0	$\sqrt{2}$	2	2	$\sqrt{2}$	2	$\sqrt{6}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{8}$
12	$\sqrt{6}$	2	$\sqrt{2}$	2	2	$\sqrt{2}$	0	$\sqrt{2}$	$\sqrt{6}$	2	$\sqrt{2}$	2	$\sqrt{8}$	$\sqrt{6}$	2	$\sqrt{6}$
13	2	$\sqrt{6}$	2	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{2}$	0	2	$\sqrt{6}$	2	$\sqrt{2}$	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	2
20	2	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{2}$	2	$\sqrt{6}$	2	0	$\sqrt{2}$	2	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{6}$	2
21	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{8}$	2	$\sqrt{2}$	2	$\sqrt{6}$	$\sqrt{2}$	0	$\sqrt{2}$	2	2	$\sqrt{2}$	2	$\sqrt{6}$
22	$\sqrt{8}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{6}$	2	$\sqrt{2}$	2	2	$\sqrt{2}$	0	$\sqrt{2}$	$\sqrt{6}$	2	$\sqrt{2}$	2
23	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	2	2	$\sqrt{6}$	2	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{2}$	0	2	$\sqrt{6}$	2	$\sqrt{2}$
30	$\sqrt{2}$	2	$\sqrt{6}$	2	2	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{2}$	2	$\sqrt{6}$	2	0	$\sqrt{2}$	2	$\sqrt{2}$
31	2	$\sqrt{2}$	2	$\sqrt{6}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{8}$	2	$\sqrt{2}$	2	$\sqrt{6}$	$\sqrt{2}$	0	$\sqrt{2}$	2
32	$\sqrt{6}$	2	$\sqrt{2}$	2	$\sqrt{8}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{6}$	2	$\sqrt{2}$	2	2	$\sqrt{2}$	0	$\sqrt{2}$
33	2	$\sqrt{6}$	2	$\sqrt{2}$	$\sqrt{6}$	$\sqrt{8}$	$\sqrt{6}$	2	2	$\sqrt{6}$	2	$\sqrt{2}$	$\sqrt{2}$	2	$\sqrt{2}$	0

Proceeding further, for signal sequences of length 3,

$$\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\mathbf{S}} = \begin{bmatrix} 0(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & 2(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} \\ \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & 0(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & 2(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} \\ 2(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & 0(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} \\ \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & 2(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & \sqrt{2}(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} & 0(Ed)\mathbf{d}_{\mathbf{S}\times\mathbf{S}} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \sqrt{2} & 2 & \sqrt{2} \\ \sqrt{2} & 0 & \sqrt{2} & 2 \\ 2 & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix} (Ed) \begin{bmatrix} 0 & \sqrt{2} & 2 & \sqrt{2} \\ \sqrt{2} & 0 & \sqrt{2} & 2 \\ 2 & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix} (Ed) \begin{bmatrix} 0 & \sqrt{2} & 2 & \sqrt{2} \\ \sqrt{2} & 0 & \sqrt{2} & 2 \\ 2 & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 2 & \sqrt{2} & 0 \end{bmatrix}.$$

2.4 Signal Constellations

In this chapter, nothing has been mentioned about the type of the signal constellations. The discussions and the representations are general and can be employed, in fact, for any signal

constellations. This thesis basically refers to three types of channel signal constellations:

(1) Actual channel signal constellations,

(2) Virtual channel signal constellations and

(3) Arbitrary channel signal constellations.

An **actual channel signal constellation** is the signal constellation whose signals are actually transmitted physically over the channel. An **virtual channel signal constellation** is a hypothetical signal constellation, which is, in fact, a block code. This signal constellation, in turn, uses an actual channel signal constellation. The general signal constellation which may be actual or virtual is termed as **arbitrary channel signal constellation.** An arbitrary channel signal constellation may be a PSK, ASK, QAM, or any other signal constellation in which the number of signals can be any number (not necessarily a prime or a power of a prime number). The channel signal constellation of a particular type is selected depending on the application and is required for the purpose of coding. It is to be understood that when a virtual channel signal constellation is used for coding, the signals of this signal constellation are not outputs of an actual modulator, which will be carried over the channel. It is to be noted that all the discussions carried so far in this chapter and later on in the thesis are valid for arbitrary channel signal constellations.

In the thesis the schemes presented give code words for a specified signal constellation. The problem of finding an optimum channel signal constellation for some specific application is not dealt here. It is emphasized that only the set of signals forming the signal constellation and the Euclidean distances between them are of prime importance for the encoding and the decoding process, how this signals are actually obtained is not of concern. Virtual channel signal constellations are further encountered in Chapter 7.

Example 2.4.1 Consider the following virtual channel signal constellation.

The actual channel signal constellation is 3-PSK signal constellation given in Appendix A. Using this a virtual channel signal constellation with 9 signals, which is actually a block code with $n = 3$, is obtained.
For the virtual channel signal constellation the set of signals is,

$$S' = \{ s'_0, \ s'_1, \ s'_2, \ s'_3, \ s'_4, \ s'_5, \ s'_6, \ s'_7, \ s'_8 \}.$$

Where the virtual signals are block code words of 3-PSK of length 3,
$s'_0 = 002$, $s'_1 = 010$, $s'_2 = 021$, $s'_3 = 100$,
$s'_4 = 111$, $s'_5 = 122$, $s'_6 = 201$, $s'_7 = 212$, $s'_8 = 220$.
Calculating the Euclidean distances between the signals of the virtual channel signal constellation, the set S is as follows,
$S = \{$ $(s'_0, s'_0, 0)$, $(s'_0, s'_1, \sqrt{6})$, $(s'_0, s'_2, \sqrt{6})$, $(s'_0, s'_3, \sqrt{6})$, $(s'_0, s'_4, 3)$, $(s'_0, s'_5, \sqrt{6})$,
$(s'_0, s'_6, \sqrt{6})$, $(s'_0, s'_7, \sqrt{6})$, $(s'_0, s'_8, 3)$, $(s'_1, s'_0, \sqrt{6})$, $(s'_1, s'_1, 0)$, $(s'_1, s'_2, \sqrt{6})$,
$(s'_1, s'_3, \sqrt{6})$, $(s'_1, s'_4, \sqrt{6})$, $(s'_1, s'_5, 3)$, $(s'_1, s'_6, 3)$, $(s'_1, s'_7, \sqrt{6})$, $(s'_1, s'_8, \sqrt{6})$,
$(s'_2, s'_0, \sqrt{6})$, $(s'_2, s'_1, \sqrt{6})$, $(s'_2, s'_2, 0)$, $(s'_2, s'_3, 3)$, $(s'_2, s'_4, \sqrt{6})$, $(s'_2, s'_5, \sqrt{6})$,
$(s'_2, s'_6, \sqrt{6})$, $(s'_2, s'_7, 3)$, $(s'_2, s'_8, \sqrt{6})$, $(s'_3, s'_0, \sqrt{6})$, $(s'_3, s'_1, \sqrt{6})$, $(s'_3, s'_2, 3)$,
$(s'_3, s'_3, 0)$, $(s'_3, s'_4, \sqrt{6})$, $(s'_3, s'_5, \sqrt{6})$, $(s'_3, s'_6, \sqrt{6})$, $(s'_3, s'_7, 3)$, $(s'_3, s'_8, \sqrt{6})$,
$(s'_4, s'_0, 3)$, $(s'_4, s'_1, \sqrt{6})$, $(s'_4, s'_2, \sqrt{6})$, $(s'_4, s'_3, \sqrt{6})$, $(s'_4, s'_4, 0)$, $(s'_4, s'_5, \sqrt{6})$,
$(s'_4, s'_6, \sqrt{6})$, $(s'_4, s'_7, \sqrt{6})$, $(s'_4, s'_8, 3)$, $(s'_5, s'_0, \sqrt{6})$, $(s'_5, s'_1, 3)$, $(s'_5, s'_2, \sqrt{6})$,
$(s'_5, s'_3, \sqrt{6})$, $(s'_5, s'_4, \sqrt{6})$, $(s'_5, s'_5, 0)$, $(s'_5, s'_6, 3)$, $(s'_5, s'_7, \sqrt{6})$, $(s'_5, s'_8, \sqrt{6})$,
$(s'_6, s'_0, \sqrt{6})$, $(s'_6, s'_1, 3)$, $(s'_6, s'_2, \sqrt{6})$, $(s'_6, s'_3, \sqrt{6})$, $(s'_6, s'_4, \sqrt{6})$, $(s'_6, s'_5, 3)$,
$(s'_6, s'_6, 0)$, $(s'_6, s'_7, \sqrt{6})$, $(s'_6, s'_8, \sqrt{6})$, $(s'_7, s'_0, \sqrt{6})$, $(s'_7, s'_1, \sqrt{6})$, $(s'_7, s'_2, 3)$,
$(s'_7, s'_3, 3)$, $(s'_7, s'_4, \sqrt{6})$, $(s'_7, s'_5, \sqrt{6})$, $(s'_7, s'_6, \sqrt{6})$, $(s'_7, s'_7, 0)$, $(s'_7, s'_8, \sqrt{6})$,
$(s'_8, s'_0, 3)$, $(s'_8, s'_1, \sqrt{6})$, $(s'_8, s'_2, \sqrt{6})$, $(s'_8, s'_3, \sqrt{6})$, $(s'_8, s'_4, 3)$, $(s'_8, s'_5, \sqrt{6})$,
$(s'_8, s'_6, \sqrt{6})$, $(s'_8, s'_7, \sqrt{6})$, $(s'_8, s'_8, 0)$, $\}$.
The table of Euclidean distances between signals obtained from the set S is as shown in Table 2.4.1 The matrix of Euclidean distances is as follows,

$$\mathbf{d_S} = \begin{bmatrix} 0 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 \\ \sqrt{6} & 0 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 & 3 & \sqrt{6} & \sqrt{6} \\ \sqrt{6} & \sqrt{6} & 0 & 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 & \sqrt{6} \\ \sqrt{6} & \sqrt{6} & 3 & 0 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 & \sqrt{6} \\ 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 0 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 \\ \sqrt{6} & 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 0 & 3 & \sqrt{6} & \sqrt{6} \\ \sqrt{6} & 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 & 0 & \sqrt{6} & \sqrt{6} \\ \sqrt{6} & \sqrt{6} & 3 & 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 0 & \sqrt{6} \\ 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 3 & \sqrt{6} & \sqrt{6} & \sqrt{6} & 0 \end{bmatrix}.$$

Table 2.4.1: Euclidean distances between signals of virtual channel signal constellation of Example 2.4.1

d	s'_0	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8
s'_0	0	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3
s'_1	$\sqrt{6}$	0	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3	3	$\sqrt{6}$	$\sqrt{6}$
s'_2	$\sqrt{6}$	$\sqrt{6}$	0	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3	$\sqrt{6}$
s'_3	$\sqrt{6}$	$\sqrt{6}$	3	0	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3	$\sqrt{6}$
s'_4	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	0	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3
s'_5	$\sqrt{6}$	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	0	3	$\sqrt{6}$	$\sqrt{6}$
s'_6	$\sqrt{6}$	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3	0	$\sqrt{6}$	$\sqrt{6}$
s'_7	$\sqrt{6}$	$\sqrt{6}$	3	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	0	$\sqrt{6}$
s'_8	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	3	$\sqrt{6}$	$\sqrt{6}$	$\sqrt{6}$	0

2.5 Concluding Remarks

This chapter introduced a set-theoretic framework for representation of signals from an arbitrary channel signal constellation and for sequences of signals of finite length. For working with various Euclidean distances, a matrix theoretic framework has been used and the Euclidean distances between the signals of a channel signal constellation are represented in the form of a matrix. The representations developed here are used in the development of efficient implementation schemes for the algorithms proposed later on in the thesis.

Chapter 3

Structured Distance Approach to Block Coded Modulation

3.1 Introduction

Coded modulation schemes which use a block encoder are known as block coded modulation [BCM] schemes. The objective of this chapter is to develop a framework to search for codes to be used with BCM schemes. Implementation of block encoders for general codes is also discussed. Various schemes for obtaining codes for BCM, summarized in Chapter 1 are limited to certain types of codes and signal constellations. Compared to these, the proposed framework is more general, in the sense that, it can be used with arbitrary channel signal constellations and general (non-linear) codes can be obtained. These generalizations help in the identification of various trade-offs which can be fruitfully employed in the selection of a particular scheme for some applications.

The chapter begins with a brief overview of some essential terminology, preliminary notations and definitions. With this background a different view to the problem of obtaining codes for BCM, known as **the structured distance approach** is presented. Subsequently, the properties of the codes obtained by this scheme and some block encoding schemes are explained. An algorithm summarizes this approach and some examples illustrate it. Finally, the chapter ends with some concluding remarks. This chapter is concerned only with the code search and encoding for the codes to be used with BCM. The soft decoding of these codes is dealt with in Chapters 5 and 6.

3.2 Background and Preliminaries

Traditionally, channel coding is performed assuming a binary symmetric channel and using the Hamming distance metric. Redundancy is added in time and coding results in a reduction of the transmission rate.

Coded modulation deals with band-limited, power-limited channels and assumes a discrete input, analog output channel. If the channel is additive white Gaussian noise [AWGN], coding is performed using the Euclidean distance metric. Redundancy can be added in the signal space or in space and time. These codes are soft decoded and can be considered to be more general than the traditional channel codes. Addition of redundancy in space is achieved by using an expanded channel signal constellation which has more signals. In this thesis, schemes are developed which assume that the expanded channel signal constellation is an arbitrary channel signal constellation as defined in Chapter 2.

Block coded modulation [BCM] schemes use block codes of a finite block length n, obtained from channel signals belonging to an expanded channel signal constellation. These codes are generated by a block encoder and are soft decoded. Compared to trellis coded modulation [TCM], BCM uses codes of finite short block lengths.

The block codes used in BCM schemes can be of a variety of types. Some of these are listed here.

Linear code: A linear code is one whose code words are generated from the data word by a linear mapping, $V_m = U_m G$ where $m = 1,2, \ldots n$, n is the block length and G is an invertible matrix of linear transformations [86].

Non-linear code: A code which is not a linear code is known as a non-linear code.

Cyclic code: A code is cyclic if the cyclic shift of a code word is also a code word [58].

Lattice code: A lattice Λ is an infinite discrete subset of $\mathbf{R}^N$ that forms an additive group under ordinary vector addition. A lattice code is a finite subset of points from a lattice Λ, or from a translate $\Lambda + \tau$ [28].

Group code: A group sequence space is a direct product $W = \prod_{k \in I} G_k$, where the time axis I is any subset of the integers Z, and the symbol alphabets G_k, $k \in I$, are

arbitrary groups. A group code is any subgroup C of a group sequence space. If all symbol alphabets G_k are equal to a common group G, then the sequence space is denoted by $W = G^I$, and C is called a group code over G defined on I. A linear code over a field, vector space, ring or module A is, at a more primitive level, a group code over the additive group of A. Thus every conventional linear code (block or lattice) is a group code [29].

Geometrically uniform code: A geometrically uniform [GU] code is a finite or infinite set of points C in a finite or infinite dimensional Euclidean space having a transitive symmetry group, that is, for any two points s_1 and $s_2 \in C$, there exists an isometry which transforms s_1 to s_2 while leaving C invariant [28]. The Voronoi regions [21] of GU codes are congruent, which essentially means that their Euclidean distance and error probability can be determined by assuming that the encoded all-zero sequence is a code word [9]. GU codes are more general than group codes and can be linear or non-linear codes.

Rectangular code: A code C of block length $n > 1$ is said to be rectangular code, if. for all $t \in [1, n-1]$, $\{ ac, ad, bc \} \subset C$ implies $bd \in C$, for all possible choices of $a, b \in P_t(C)$ and $c, d \in F_t(C)$, where the code C has past $P_t(C)$ and future $F_t(C)$ at depth t. Rectangular codes can be linear or non-linear [49].

General code: A code which can be of any type is termed as a general code [12].

This thesis deals with general codes, which is the widest class of block codes. This class includes the non-linear codes. The motivation for considering non-linear codes is that for a small block length n, for a prescribed minimum Euclidean distance $d_{\min}$, sometimes, this class of codes result in more number of code words and hence a larger rate. For coded modulation non-linear codes are also required as they are more suitable for taking care of the effects of the carrier-phase offsets [80].

The main disadvantage of codes without any inherent structure is that a compact representation of the codes in terms of a reduced basis is not possible. The full code table is required by the encoder and decoder. With the evolution of technology and the easy availability of dense memory chips, this no longer seems to be an unsurmountable problem.

Non-linear codes can be considered to trade off memory for more number of code words which gives a better rate.

3.3 Codes for Block Coded Modulation

A block coded modulation [BCM] scheme uses block codes which are obtained from the signals of an expanded channel signal constellation. The code words are separated by an Euclidean distance which is greater than the Euclidean distance for the uncoded case. This is specified in terms of the coding gain. The BCM scheme also has a rate, which can be compared to the rate of the uncoded scheme. These terms are defined and explained in this section.

3.3.1 Block Code

If it is required to transmit information over a channel at rate B_1 symbols/s. The channel provides an bandwidth for transmission at rate B_2 symbols/s. Then a channel signal constellation provided by the modulator with n'_b signal symbols will be required.

Definition 3.3.1 *This channel signal constellation B' with n'_b signals, used for uncoded communication, is known as the* **base signal constellation.**

If coded modulation is used, then redundancy has to be added and a channel signal constellation with a larger signal space is required.

Definition 3.3.2 *The channel signal constellation S' with n' signals such that $n' > n'_b$, used with coded modulation is known as the* **expanded signal channel constellation.**

The objective of the code design is to make the minimum Euclidean distance between the code words $d_{\min}$ larger than the Euclidean distance for uncoded modulation case d_{uc}. Consider an expanded channel signal constellation S' as defined in Section 2.2.

Definition 3.3.3 *If $s'_{i_1} s'_{i_2} \ldots s'_{i(n-1)} \in C$ and $s'_{j_1} s'_{j_2} \ldots s'_{j(n-1)} \in C$ are two code words. Then,*

$$d_{\min} = \min \sqrt{(s'_{i_1} - s'_{j_1})^2 + \ldots + (s'_{i_{n-1}} - s'_{j_{n-1}})^2} \neq 0, \quad \forall i, j \ s.t. \ s'_i \ and \ s'_j \in C.$$

Definition 3.3.4 *d_{uc} is the minimum Euclidean distance (non-zero) between the signals of the base signal constellation B'.*

The channel is assumed to be power-limited and hence no change in the power level at transmission is possible.

Definition 3.3.5 *A block code of block length n, for a BCM scheme is defined to be the set of code words C, such that $C \subset S' \times S' \times \ldots \times S'_{(n\text{-times})}$ where, $S' \times S' \times \ldots \times S'_{(n\text{-times})}$ is the set of all the sequences of length n of signals from the expanded channel signal constellation.*

The block code for a BCM scheme is represented as (B', S', n, $|C|$, $d_{\min}$).
Where,
B' – is the base signal constellation,
S' – is the expanded channel signal constellation,
n – is the block length,
$|C|$ – is the number of code words and
$d_{\min}$ – is the minimum Euclidean distance between the code words.

3.3.2 Equivalent Block Codes

Two codes are equivalent if they differ only in the order of the symbols [58]. These are of interest, mainly from the soft decoding point of view as discussed in Section 5.5.2.

Definition 3.3.6 *Consider two codes C_1 and C_2, of block length n, with signals from some expanded channel signal constellation S'.*
C_1 is equivalent to C_2 if there exists a permutation α of the n co-ordinate positions such that,

$$\text{if, } (s'_{i_1}, s'_{i_2}, \ldots, s'_{i_n}) \in C_1, \text{ then } \alpha(s'_{i_1}, s'_{i_2}, \ldots, s'_{i_n}) \in C_2.$$

3.3.3 Asymptotic Coding Gain

Assuming AWGN channel and soft maximum likelihood decoding, the coding results in an improvement in noise immunity, due to increase in $d_{\min}$ over d_{uc}. This is specified as the coding gain.

R can also be called the bandwidth utilization factor [BUF]. It specifies the efficiency of the coded modulation scheme in utilization of the available bandwidth.

Example 3.3.1 Consider the example of the following block code used with an BCM scheme.

(2-PSK, 5-PSK, 3, 10, $\sqrt{5}$) is a block code for a BCM scheme.
The base signal constellation is the 2-PSK signal constellation, shown in Appendix A.
The expanded channel signal constellation is the 5-PSK signal constellation, shown in Appendix A.
The block length is 3.
The 10 code words are,
$C = \{000, 012, 024, 031, 043, 201, 213, 220, 232, 244\}$.
There are $2^3 = 8$ data words, $\{000, 001, 010, 011, 100, 101, 110, 111\}$.
The $d_{\min}$ for the code words $= \sqrt{5}$ and d_{uc} for 2-PSK $= 2$.
For this code,

$$G = 10\log_{10}(5/4) = 0.96\ dB \text{ and}$$

$$R = \left(\frac{\log_2 10}{3}\right) = 1.1.$$

$R > 1$, as there are more code words (10), and less data words (8).
This BCM scheme can also be used to transmit another set of 10 data words.

Now consider the following (2-PSK, 5-PSK, 3, 7, $\sqrt{6.38}$) block code for a BCM scheme.
The base signal constellation is the 2-PSK signal constellation, shown in Appendix A.
The expanded channel signal constellation is the 5-PSK signal constellation, shown in Appendix A.
The block length is 3.
The 7 code words are,
$C = \{000, 022, 134, 202, 314, 331, 443\}$.
There are $2^3 = 8$ data words, $\{000, 001, 010, 011, 100, 101, 110, 111\}$. The $d_{\min}$ for the code words $= \sqrt{6.38}$ and d_{uc} for 2-PSK $= 2$.
For this code,

$$G = 10\log_{10}(6.38/4) = 2.03\ dB \text{ and}$$

$$R = \left(\frac{\log_2 7}{3}\right) = 0.94.$$

$R < 1$, as there are less code words (7), and more data words (8).
This BCM scheme can be used to transmit a set of 7 data words.

3.4 Code Search

A brute force approach to the problem of finding general codes for BCM schemes with arbitrary channel signal constellations under this general frame work, will involve searching Euclidean distance matrices between signals and Euclidean distance distribution tables between signal sequences of finite length. With the existing computational technologies such a search can be carried out for obtaining codes. The complexity and time required for the search increases with increasing n and n'.

BCM schemes have to be developed using codes suitable for various specific applications. Another important issue in obtaining the codes for a BCM scheme is the soft decoding of these codes. The codes obtained should be such that practical implementation of soft decoders is possible.

As more and more structure is added to the codes, the code search becomes easier. For example, consider GU codes [28], which include a class of linear and non-linear codes. Due to the symmetry of these codes, the search for the codes can start from any point, that is, with any initial code word. In the case of group codes [29], the search is limited to sub-groups of certain groups. Linear cyclic codes [42] have such a rich structure that the search and implementation issues are simplified. And, various authors have presented various schemes which are summarized in Chapter 1.

In this thesis, with the evolving technological changes of the future in mind, a general framework is presented which provides trade offs in the selection of codes for BCM schemes. These basically encompass a wider range of applications and results in a finer choice in the selection and implementation of codes for BCM schemes.

3.5 Motivating Factors

Example 3.5.1 Example illustrating the importance of an arbitrary channel signal constellation.

Consider the 4-PSK signal constellation, shown in Appendix A.
This is used as the expanded signal constellation for a BCM scheme and the base signal constellation is 2-PSK with $d_{\text{uc}} = 2$.
If a code is required for a BCM scheme with block length $n = 2$, and $d_{\text{min}} > 2$, then only two code words are possible and $C = \{00, 22\}$.
This gives the code, (2-PSK, 4-PSK, 2, 2, $\sqrt{8}$).
Now consider the asymmetric 4-PSK signal constellation, shown in Appendix A.
This is used as the expanded channel signal constellation for a BCM scheme and the base signal constellation is 2-PSK with $d_{\text{uc}} = 2$.
If a code is required for a BCM scheme with block length $n = 2$, and $d_{\text{min}} > 2$, then four code words are possible $C = \{00, 12, 21, 33\}$.
This gives the code, (2-PSK, Asy 4-PSK, 2, 4, $\sqrt{5}$).
So in this case, for the same coding problem, an asymmetric channel signal constellation gives more code words than a symmetric channel signal constellation.

FACT − 1: The use of an asymmetric expanded channel signal constellation can result in more number of code words than possible by the use of a symmetric expanded channel signal constellation.

Example 3.5.2 Example illustrating the importance of non-linear codes for BCM.

Consider the 4-PSK signal constellation, shown in Appendix A.
This is used as the expanded channel signal constellation for a BCM scheme and the base signal constellation is 2-PSK with $d_{\text{uc}} = 2$.
If a code is required for a BCM scheme with block length $n = 4$, and $d_{\text{min}} > \sqrt{6}$, then only 16 code words exist, if attention is restricted to linear codes.
For example,
$C = \{$0000, 0022, 2002, 2200, 0220, 0202, 2020, 2222, 1111, 1133, 3113, 3311, 1331, 1313,

3131 3333 }.
This gives the linear code, (2-PSK, 4-PSK, 4, 16, $\sqrt{8}$).
Now if the restriction of linearity on the code words is removed, 19 code words result, as follows,
{ 0000, 0021, 0123, 0230, 0212, 0332, 1002, 1110, 1131, 1203, 1311, 1323, 2013, 2030, 2122. 2201, 2220, 3102, 3303 }.
This gives the non-linear code, (2-PSK, 4-PSK, 4, 19, $\sqrt{6}$)[1].
So in this case, for the same coding problem, a non-linear code has more code words than any possible linear code.

FACT – 2: Non-linear codes can have more code words than any possible linear codes for some BCM applications.

Example 3.5.3 Example illustrating the use of an expanded channel signal constellation that need not have double the number of signals present in the base signal constellation.

Consider the the 4-PSK signal constellation, shown in Appendix A.
This is used as the expanded channel signal constellation for a BCM scheme and the base signal constellation is 2-PSK with $d_{\text{uc}} = 2$.
Consider the code for a BCM scheme with block length $n = 3$, and $d_{\min} = \sqrt{6}$.
A maximum of 8 code words is possible. For example consider the following code.
$C = \{ 000, 022, 202, 220, 113, 131, 311, 333 \}$.
This is the linear code, (2-PSK, 4-PSK, 3, 8, $\sqrt{6}$).
No other code with $|C| > 8$ for this channel signal constellation exists .
Now consider the expanded channel signal constellation to be the 3-PSK signal constellation. illustrated in Appendix A.
For the same BCM scheme, that is with $n = 3$ and $d_{\min} = \sqrt{6}$, 9 code words exist.
{ 001, 012, 020, 100, 111, 122, 202, 210, 221 }.
This gives the code, (2-PSK, 3-PSK, 3, 9, $\sqrt{6}$).
So in this case, for the same coding problem, an expanded channel signal constellation with less number of signals than a constellation with double the number of signals from the number of signals in the base signal constellation, gives more number of code words.

[1]This is not a linear, cyclic, group, lattice, GU, or rectangular code.

FACT – 3: The expanded channel signal constellation need not have double the number of signals as contained in the base signal constellation. The number of signals can be more or less than double, but the inequality in Definition 3.3.2 has to be satisfied.

These factors have motivated the nature of the search for finding block codes to be used with BCM schemes. The encoding and soft-decoding schemes discussed in this thesis have also been developed for handling codes obtained based on these considerations.

3.6 The Structured Distance Approach

It is possible to obtain code words for a BCM scheme by a brute force search, but, it is of interest to employ some structure in the search, so that the process is simplified.

The two basic approaches employed in the existing literature can be classified as follows.

(1) **The structured code approach:** In this approach, structure is imposed on the code words. For example, the linear codes [5, 6, 42, 63], lattice codes [8, 17, 26, 31, 53], group codes [19,29,33,57,83], rectangular codes [49,82] and GU codes [28] are all codes based on this approach.

(2) **The structured encoder approach:** In this approach, structure is imposed on the encoder which generates the code words. For example the multilevel codes discussed in [41, 46].

This section proposes a new approach known as **the structured distance approach.** First consider the following observations:

- Coded modulation over AWGN channels uses the Euclidean distance metric as against the Hamming distance metric used in the conventional codes. The codes have to be designed to maximize the minimum Euclidean distance between the code words and not the Hamming distance. For a general channel signal constellation, which has more than two signals in a modulation dimension, the Hamming distance and the Euclidean distance are not equivalent.

- In conventional codes since the Hamming distance metric is used, the code words (mapped to labels) have the information of the distance. The distance between two code words is calculated from the code words themselves. But, this is not so, for the case of Euclidean distances. In general, for an arbitrary channel signal constellation it might not be possible to map the signals with labels such that the codewords have the information of the Euclidean distances. This information has to be obtained from the matrix of the Euclidean distances associated with the channel signal constellation.

- For a channel signal constellation, generally, the number of elements in the set of Euclidean distances between the signals is much less than the number of signals in the channel signal constellation.[2] For example the 8-PSK signal constellation has 8 signals but only 3 Euclidean distances and the 16 QAM has 16 signals but only 9 Euclidean distances (The zero distance element in the set is not considered). This is mainly due to the inherent symmetries in the channel signal constellations used with the various modulation techniques.

- The encoder generates the code words, but the coding problem is to obtain optimum codes for certain applications. The main aim is to maximize the minimum Euclidean distances. For a general problem using an arbitrary channel signal constellation, it cannot be always assured that a particular system will result in the maximization of the minimum Euclidean distances between the code words.

These observations primarily motivate for a change of viewpoint to the coding problem. The new point of view results in **the structured distance approach.** The salient features of the structured distance approach are as follows.

(1) No structure is assumed on the code words.

(2) No specific type of encoder is assumed before obtaining the codes.

(3) No specific structure is assumed for the channel signal constellation.

(4) The Euclidean distances between the signals provided by an expanded channel signal constellation are used.

[2]Refer Table 2.2.2.

(5) The set of Euclidean distances provided by an expanded channel signal constellation and the $d_{\min}$ required for the BCM scheme structures the manifestation of the Euclidean distance distribution for the code.

(6) Code search consists of obtaining code words such that the minimum Euclidean distance between all the codes is $\geq d_{\min}$.

(7) The Euclidean distances are the prime entities and no consideration is given to the soft-decoding complexity of the code.

This approach shifts the whole perspective from the code words or sequences of signals of a channel signal constellation to the Euclidean distance distribution or sequences of Euclidean distances between signals provided by the channel signal constellation. Irrespective of what the code words may be, coding has to just assure that all the elements in the distance distribution are $\geq d_{\min}$. The generality of this approach is evident. Linear codes, lattice codes, group codes, GU codes and rectangular codes can be viewed as certain codes or classes of codes which have certain types of structure on the Euclidean distance distribution, for certain specific channel signal constellations and specific mappings between distances and signals. In this approach, first, the code words are obtained and then an encoder and a soft decoder for the code has to be designed. In this section, the framework required for working with Euclidean distances is developed and a scheme for obtaining codes is given.

Definition 3.6.1 *Given a channel signal constellation with n' signals $\{s'_0, s'_1, \ldots, s'_{(n'-1)}\}$, the set of all the Euclidean distances between signals of this channel signal constellation is,*

$$D = \{0, d_1, d_2, \ldots, d_p\}, \text{ where } p \leq \frac{(n'-1)n'}{2}.$$

The d_k's of set D are, in fact, the elements $d_{i,j}$'s of the matrix $\mathbf{d_S}$ discussed in Section 2.2. Here a different notation is required as now a representation of distance, independent of the signals is necessary.

Definition 3.6.2 *On the set D, a binary operation* **distance composition** *denoted by the symbol '$\circ$', is defined as follows.*

$$\forall i, j, k, \;\; s.t. \; d_i, d_j, d_k \in D,$$

$$d_i \circ d_j \in D,$$

$$d_i \circ d_j = d_j \circ d_i,$$

$$\exists 0 \in D \ s.t. \ 0 \circ d_i = d_i \circ 0 = d_i \ and$$

$$d_i \circ (d_j \circ d_k) = (d_i \circ d_j) \circ d_k.$$

The significance of the operation of composition of distance '$\circ$' is that, if d_i is the Euclidean distance between two signals s'_0 and s'_1, and d_j is the Euclidean distance between two signals s'_1 and s'_2, then the Euclidean distance between the signals s'_0 and s'_2 will be $d_i \circ d_j$.

Theorem 3.6.1 *$\{D, \circ\}$ is a monoid.*

Proof: From Definition 3.6.2, $\{D, \circ\}$ satisfies the associativity law since,

$$d_i \circ (d_j \circ d_k) = (d_i \circ d_j) \circ d_k.$$

D has an identity element 0 since,

$$\exists 0 \in D \text{ s.t. } 0 \circ d_i = d_i \circ 0 = d_i.$$

Hence, $\{D, \circ\}$ is a monoid. □

Theorem 3.6.2 *For an arbitrary channel signal constellation, $d_i \circ d_j$ need not correspond to a single element of D, but in general $d_i \circ d_j \subset D$.*

Proof: **CASE 1:** Assume that $d_i = d_j$.
The Euclidean distance between signals is a scalar quantity.
For an arbitrary channel signal constellation, the signals of the channel signal constellation are in the Euclidean space R^q depending on the dimensionality q of the channel signal constellation.
Depending on the direction in which the distances are composed, one possibility is that,

$$d_i \circ d_i = d_k \text{ s.t. } d_k \in D.$$

For a different direction the composition may result in,

$$d_i \circ d_i = 0.$$

Hence,

$$d_i \circ d_i = \{0, d_k\} \subset D.$$

CASE 2: Assume that $d_i \neq d_j$.
In this case depending on the direction in which the distances are composed,

$$d_i \circ d_j = d_{k1}, \text{ s.t. } d_{k1} \in D \text{ or } d_i \circ d_j = d_{k2}, \text{ s.t. } d_{k2} \in D.$$

Hence,

$$d_i \circ d_j = \{d_{k1}, d_{k2}\} \subset D.$$

So, for an arbitrary channel signal constellation, $d_i \circ d_j$ need not correspon(element of D, but in general $d_i \circ d_j \subset D$. □

Example 3.6.1 Consider the 4-PSK signal constellation discussed in Appendi

For this channel signal constellation, $D = \{0, \sqrt{2}, 2\}$.
Now consider the Euclidean distance $\sqrt{2}$.

$$\sqrt{2} \circ \sqrt{2} = \{0, 2\} \subset D.$$

That is, consider the two signals of the channel signal constellation s'_0 and s'_1 separated by the Euclidean distance $\sqrt{2}$. From s'_1, if

another signal is separated by the Euclidean distance $\sqrt{2}$, then the new si(either s'_0 or it can be s'_2. These are at a Euclidean distance of 0 or 2, respectivel To overcome this problem the power set of D is defined as follows.

Definition 3.6.3 *The power set of the set D, is denoted by P(D),*

$$P(D) = \{\emptyset, \{0\}, \{d_1\}, \ldots, \{0, d_1, \ldots, d_p\}\}.$$

Definition 3.6.4 *The composition operation between sets of distances is defined*

$$\{d_i, d_j\} \circ \{d_k, d_l\} = \{d_i \circ d_k, d_i \circ d_l, d_j \circ d_k, d_j \circ d_l\} \text{ and}$$

$$\emptyset \circ d_i = d_i \circ \emptyset = \emptyset \circ \emptyset = \emptyset.$$

Theorem 3.6.3 $\{P(D), \circ\}$ *is a monoid.*

Proof: From Definitions 3.6.2 and 3.6.4, $\{P(D), \circ\}$ satisfies the associativity law. $P(D)$ has an identity element $\{0\}$. Hence, $\{P(D), \circ\}$ is a monoid. □

Example 3.6.2 Consider the 2-PSK signal constellation shown in Appendix A.

The set of Euclidean distances $D = \{0,\ d_1\}$, where $d_1 = 2$.
In this case it is not necessary to define $P(D)$, as composition cannot lead to a set of distances. The table for the composition operation is shown in Table 3.6.1. Now consider the

Table 3.6.1: Distance composition for the 2-PSK signal constellation

$\circ$	0	d_1
0	0	d_1
d_1	d_1	0

4-PSK signal constellation shown in Appendix A.
The Euclidean distances between the signals are denoted by, $d_1 = \sqrt{2}$ and $d_2 = 2$.
The set of Euclidean distances $D = \{0,\ \sqrt{2},\ 2\} = \{0,\ d_1,\ d_2\}$.

$$P(D) = \{\ \emptyset, \{0\}, \{d_1\}, \{d_2\}, \{0, d_1\}, \{0, d_2\}, \{d_1, d_2\}, \{0, d_1, d_2\}\ \}$$

The Table 3.6.2 gives the composition operation on this power set $P(D)$. The entries corresponding to the subset $\{\{0\},\ \{d_1\},\ \{d_2\}\}$ are of particular interest. It is to be noted that it is not always necessary to consider the power set of distances, for practical purposes just checking that a distance obtained is valid for the distance distribution of the code is enough.

Definition 3.6.5 *For a channel signal constellation S' with the set of Euclidean distances $D = \{0,\ d_1,\ \ldots,\ d_p\}$, the set of the n-tuples of distances is,*

$$D_{S'\times S'\times\ldots\times S'\,(n\text{-times})} = \{\,00\ldots0,\ 00\ldots d_1,\ \ldots,\ d_p d_p \ldots d_p\,\}.$$

Definition 3.6.6 *For n-tuples of distances,*

$$d_{i_1} d_{i_2} \ldots d_{i_n} \in D_{S'\times S'\times\ldots\times S'\,(n\text{-times})} \text{ and } d_{j_1} d_{j_2} \ldots d_{j_n} \in D_{S'\times S'\times\ldots\times S'\,(n\text{-times})},$$

$$d_{i_1} d_{i_2} \ldots d_{i_n} \circ d_{j_1} d_{j_2} \ldots d_{j_n} = d_{i_1} \circ d_{j_1} d_{i_2} \circ d_{j_2} \ldots d_{i_n} \circ d_{j_n} \in D_{S'\times S'\times\ldots\times S'\,(n\text{-times})}.$$

Table 3.6.2: Distance composition for the 4–PSK signal constellation

$\circ$	$\emptyset$	$\{0\}$	$\{d_1\}$	$\{d_2\}$	$\{0d_1\}$	$\{0d_2\}$	$\{d_1d_2\}$	$\{0d_1d_2\}$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\{0\}$	$\emptyset$	$\{0\}$	$\{d_1\}$	$\{d_2\}$	$\{0d_1\}$	$\{0d_2\}$	$\{d_1d_2\}$	$\{0d_1d_2\}$
$\{d_1\}$	$\emptyset$	$\{d_1\}$	$\{0d_2\}$	$\{d_1\}$	$\{0d_1d_2\}$	$\{d_1\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$
$\{d_2\}$	$\emptyset$	$\{d_2\}$	$\{d_1\}$	$\{0\}$	$\{d_1d_2\}$	$\{0d_2\}$	$\{0d_1\}$	$\{0d_1d_2\}$
$\{0d_1\}$	$\emptyset$	$\{0d_1\}$	$\{0d_1d_2\}$	$\{d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$
$\{0d_2\}$	$\emptyset$	$\{0d_2\}$	$\{d_1\}$	$\{0d_2\}$	$\{0d_1d_2\}$	$\{0d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$
$\{d_1d_2\}$	$\emptyset$	$\{d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$
$\{0d_1d_2\}$	$\emptyset$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$	$\{0d_1d_2\}$

The notation defined in this section differs from that of Chapter 2. The operation and the sets defined in this section are for Euclidean distances only without considering the signals of the channel signal constellation.

Definition 3.6.7 *For a block code of length n used for BCM, the Euclidean distance between all the code words can be represented by the n-tuples of distances from the set D of the expanded channel signal constellation.*
Each n-tuple of the distance is an element of the Euclidean distance distribution of the code. For a block code each and every element $d_{i_1}d_{i_2}\ldots d_{i_n}$ of the distance distribution is such that,

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} \geq d_{\min}.$$

For a block code with $N = |C|$ code words, the distance distribution has $\{\,(N-1)+(N-2)+\ldots+1\,\}$ elements.

Example 3.6.3 Consider the 3-PSK signal constellation shown in Appendix A.

For this channel signal constellation $D = \{\,0,\, d_1\,\}$, where $d_1 = \sqrt{3}$.
Consider a block code $C = \{\,00,\, 11,\, 22\,\}$ of block length $n = 2$ and $d_{\min} = \sqrt{6}$.
For this code the distance distribution and the code words are represented in Table 3.6.3.
Each element of the Euclidean distance distribution for the code is a 2-tuple d_1d_1.
Each element of the distance distribution is such that, $\sqrt{d_1^2 + d_1^2} = \sqrt{3+3} = \sqrt{6} = d_{\min}$.
$|C| = 3$ and the number of elements in the distance distribution is $= (2+1) = 3$.

Table 3.6.3: Code words and distance distribution for the code in Example 3.6.3

00		
	d_1d_1	
11		
	d_1d_1	d_1d_1
22		

Now the convention used for representing the distance distribution is briefly explained. The first column contains the code words.
00 and 11 are the first two code words. The distance between them is represented in the row between these code words, that is, the row above 11 with the element d_1d_1. The next code word is 22. In the row above this, the first element d_1d_1 is the distance between 11 and 22. The second element d_1d_1 is the distance between 00 and 22.
The convention for writing the distance distribution table of a code can be summarized as follows,

(1) The first column contains the code words.

(2) Code words are written on alternate rows beginning from row 1.

(3) In the row between two code words, the distance of all the code words above this row with the code word just below this row are written.

(4) The first distance in a row is the distance between the code word just below and just above this row.

(5) The distance of the code word just below the row with all the code words above the row, from the nearest code word till the first code word in the first row, are written from left to right order, in the row above a code word.

Theorem 3.6.4 *If $x_1x_2\ldots x_n$ and $y_1y_2\ldots y_n$ are code words for a BCM scheme and the Euclidean distance between them is represented by the n-tuple $d_{xy_1}d_{xy_2}\ldots d_{xy_n}$. If, $z_1z_2\ldots z_n$ is a code word such that the Euclidean distance between $y_1y_2\ldots y_n$ and $z_1z_2\ldots z_n$ is $d_{yz_1}d_{yz_2}\ldots d_{yz_n}$,*

then the Euclidean distance between $x_1x_2\ldots x_n$ *and* $z_1z_2\ldots z_n$ *is* $d_{xz_1}d_{xz_2}\ldots d_{xz_n} = (d_{xy_1} \circ d_{yz_1})\ (d_{xy_2} \circ d_{yz_2})\ \ldots\ (d_{xy_n} \circ d_{yz_n})$.

Proof: Assume that the dimensionality $n = 1$.
The signals x_1, y_1 and z_1 are seperated by distances from a set D.
$\{\, D,\ \circ \,\}$ is a monoid and Euclidean distances are transitive.
Also $(d_{xy_1} \circ d_{yz_1}) \ \in\ D$ hence, $(d_{xy_1} \circ d_{yz_1}) = d_{xz_1}$.
The code words are n dimensional and the distance between signals in each dimension is considered separately. So the above property holds for n- tuples of distances. □

Definition 3.6.8 *For a code word of block length* n, $N = |C|$ *code words, with Euclidean distance* $d_{\min}$,

$$\exists \hat{D}_{S'\times S'\times\ldots\times S'\,(n\text{-times})} \in D_{S'\times S'\times\ldots\times S'\,(n\text{-times})} \ s.t.$$

$$\forall i \ \ d_{i_1}d_{i_2}\ldots d_{i_n} \in \hat{D}_{S'\times S'\times\ldots\times S'\,(n\text{-times})}$$

$$and \ \sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} \geq d_{\min}.$$

$\hat{D}_{S'\times S'\times\ldots\times S'\,(n\text{-times})}$ *is the set of* **valid Euclidean distances** *for the distance distribution of the code.*

The distance distribution of a block code is obtained by selecting $\{\, (N-1)+(N-2)+\ldots+1 \,\}$ elements, not necessarily distinct, from the set $\hat{D}_{S'\times S'\times\ldots\times S'(n\text{-times})}$.

Some authors [42] consider the minimization of the nearest neighbor for a code word also to be an important criteria for code design. Let us consider this in terms of the distance distribution of the code.

Definition 3.6.9 *For a code with minimum Euclidean distance* $d_{\min}$, *the code words at a Euclidean distance* $d_{\min}$ *from a code word are known as the nearest neighbors for that code word. The elements of the distance distribution* $d_{i_1}d_{i_2}\ldots d_{i_n}$, *such that*

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} = d_{\min},$$

result in nearest neighbors.

Theorem 3.6.5 *For a block code of length* n *and minimum distance* $d_{\min}$, *if* $\hat{D}_{S'\times S'\times\ldots\times S'\,(n\text{-times})} = \emptyset$, *then such a coding scheme cannot exist.*

Proof: From Definition 3.6.8,

$$\hat{D}_{S' \times S' \times \ldots \times S'(n\text{-times})} = \emptyset,$$

implies that there is no element in $D_{S' \times S' \times \ldots \times S'(n\text{-times})}$ such that,

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} \geq d_{\min}.$$

So, no valid Euclidean distance exists for the distance distribution of the code. Hence, such a block code cannot exist. □

Note that if a BCM scheme is not possible, then some parameters, that is, n, $d_{\min}$ or the expanded channel signal constellation have to be changed.

Some considerations in obtaining the distance distribution and codes for BCM are to be noted.

- To obtain a distance distribution and a code satisfying it, the elements from the set $\hat{D}_{S' \times S' \times \ldots \times S'(n\text{-times})}$ can be selected (provided sufficient signals exist to satisfy the distribution and generate code words), such that the distances follow some rule, that is, some structure can be imposed on the distances.

- During the process of obtaining the distance distribution, care has to be taken, so that, distinct code words satisfying this distribution exist.

- The elements of the distance distribution need not be unique.

- In obtaining the distance distribution, every new element of the distance distribution is selected in a manner to ensure that, the composition of this new element with all the previously selected elements of the distribution are also valid Euclidean distances $\in \hat{D}_{S' \times S' \times \ldots \times S'(n\text{-times})}$.

After a distance distribution is obtained satisfying all the above mentioned constraints, signals can be assigned and code words can be obtained satisfying the distance distribution.

Example 3.6.4 Consider the 4-PSK signal constellation shown in Appendix A as the expanded channel signal constellation and 2-PSK as the base signal constellation for a BCM scheme.

Block length $n = 3$, $d_{\min} = \sqrt{6}$ and suppose that a code is required with $|C| = 8$ code words. The Euclidean distances between the signals of the 4-PSK signal constellation are denoted as $d_0 = 0$, $d_1 = \sqrt{2}$ and $d_2 = 2$.

The set of Euclidean distances is $D = \{ d_0, d_1, d_2 \}$.

For $n = 3$,

$D_{S' \times S' \times S'} = \{ d_0d_0d_0,\ d_0d_0d_1,\ d_0d_0d_2,\ d_0d_1d_0,\ d_0d_1d_1,\ d_0d_1d_2,\ d_0d_2d_0,\ d_0d_2d_1,\ d_0d_2d_2,\ d_1d_0d_0,\ d_1d_0d_1,\ d_1d_0d_2,\ d_1d_1d_0,\ d_1d_1d_1,\ d_1d_1d_2,\ d_1d_2d_0,\ d_1d_2d_1,\ d_1d_2d_2,\ d_2d_0d_0,\ d_2d_0d_1,\ d_2d_0d_2,\ d_2d_1d_0,\ d_2d_1d_1,\ d_2d_1d_2,\ d_2d_2d_0,\ d_2d_2d_1,\ d_2d_2d_2 \}$.

$|D_{S' \times S' \times S'}| = 27$.

The set of valid Euclidean distances is,

$\hat{D}_{S' \times S' \times S'} = \{ d_0d_1d_2,\ d_0d_2d_1,\ d_0d_2d_2,\ d_1d_0d_2,\ d_1d_1d_1,\ d_1d_1d_2,\ d_1d_2d_0,\ d_1d_2d_1,\ d_1d_2d_2,\ d_2d_0d_1,\ d_2d_0d_2,\ d_2d_1d_0,\ d_2d_1d_1,\ d_2d_1d_2,\ d_2d_2d_0,\ d_2d_2d_1,\ d_2d_2d_2 \}$.

Each element of $\hat{D}_{S' \times S' \times S'}$ is a Euclidean distance $\geq d_{\min}$.

Select $d_0d_1d_2 \in \hat{D}_{S' \times S' \times S'}$ as an initial distance to start constructing the distribution. This represents the Euclidean distance between two code words.

Select another distance $d_1d_2d_0$, and obtain the second row of the distribution as follows,

$d_0d_1d_2$

$d_1d_2d_0 \quad d_1 \circ d_0\ d_2 \circ d_1\ d_0 \circ d_2 = d_1d_1d_2$

The second row contains $d_1d_2d_0 \in \hat{D}_{S' \times S' \times S'}$ and the composition of this with the element in the first row.

Since the elements in the distance distribution need not be unique, again selecting $d_0d_1d_2$, the next row of distribution is obtained by composition with elements in the previous row.

$d_0d_1d_2$

$d_1d_2d_0 \quad d_1d_1d_2$

$d_0d_1d_2 \quad d_0 \circ d_1\ d_1 \circ d_2\ d_2 \circ d_0 = d_1d_1d_2 \quad d_0 \circ d_1\ d_1 \circ d_1\ d_2 \circ d_2 = d_1d_2d_0$

Note that as discussed in Table 3.6.2, $d_1 \circ d_1 = \{ d_0, d_2 \}$. But, if d_0 is selected then the last distance in the last row of the distribution discussed above becomes, $d_0 \circ d_1\ d_1 \circ d_1\ d_2 \circ d_2 = d_1d_0d_0$ and $d_1d_0d_0 \notin \hat{D}_{S' \times S' \times S'}$.

In this manner, it is not necessary to consider the power set but still by proper choice to obtain valid distances, the distribution can be obtained.

To have 8 code words there should be 7 rows in the distances distribution, these are obtained proceeding in a similar manner.

$d_0d_1d_2$
$d_1d_2d_0 \quad d_1d_1d_2$
$d_0d_1d_2 \quad d_1d_1d_2 \quad d_1d_2d_0$
$d_2d_0d_1 \quad d_2d_1d_1 \quad d_1d_1d_1 \quad d_1d_2d_1$
$d_1d_2d_0 \quad d_1d_2d_1 \quad d_1d_1d_1 \quad d_2d_1d_1 \quad d_2d_0d_1$
$d_0d_1d_2 \quad d_1d_1d_2 \quad d_1d_1d_1 \quad d_1d_2d_1 \quad d_2d_0d_1 \quad d_2d_1d_1$
$d_1d_2d_0 \quad d_1d_1d_2 \quad d_0d_1d_2 \quad d_2d_1d_1 \quad d_2d_0d_1 \quad d_1d_2d_1 \quad d_1d_1d_1$

Simultaneously with obtaining the distances of the distribution, the code words satisfying this distribution can be found. These are obtained as follows,

000
$d_0d_1d_2$
012
$d_1d_2d_0 \quad d_1d_1d_2$
132
$d_0d_1d_2 \quad d_1d_1d_2 \quad d_1d_2d_0$
120
$d_2d_0d_1 \quad d_2d_1d_1 \quad d_1d_1d_1 \quad d_1d_2d_1$
321
$d_1d_2d_0 \quad d_1d_2d_1 \quad d_1d_1d_1 \quad d_2d_1d_1 \quad d_2d_0d_1$
201
$d_0d_1d_2 \quad d_1d_1d_2 \quad d_1d_1d_1 \quad d_1d_2d_1 \quad d_2d_0d_1 \quad d_2d_1d_1$
213
$d_1d_2d_0 \quad d_1d_1d_2 \quad d_0d_1d_2 \quad d_2d_1d_1 \quad d_2d_0d_1 \quad d_1d_2d_1 \quad d_1d_1d_1$
333

The code obtained here is a linear cyclic code, (2-PSK, 4-PSK, 3, 8, $\sqrt{6}$).
This example illustrates one of the possible schemes for obtaining block codes based on the structured distance approach.

3.7 Properties of the Codes

In the structured distance approach, since Euclidean distances between signals of an expanded channel signal constellation are used as the starting point, the codes obtained are general codes. Also since the problem of finding codes for BCM has been discussed in a general manner, nothing much can be commented about the characteristics of the codes

obtained. A few properties of the codes obtained by the structured distance approach are now specified, which illustrates the generality of the approach.

Theorem 3.7.1 *The codes obtained by the structured distance approach need not be linear, cyclic, lattice, group, GU or rectangular codes.*

Proof: In the structured distance approach the codes are obtained from Euclidean distances. As the code words are not selected to satisfy any specific property, there might not be any linear transformation between the data words and the code words. Hence the codes obtained need not be linear.

Cyclic shifts of code words might not be code words. Hence the codes obtained need not be cyclic.

Also the code words need not be points on a lattice. So the codes obtained need not be lattice codes.

The set of code words might not form a sub-group. As a result of which the codes obtained need not be a group code.

GU codes are codes for which the Euclidean distances are such that, the Voronoi regions [21] for all the code words are identical and separated by just rotations and translations in the Euclidean space. A distance distribution can be selected such that this condition is satisfied. So a GU code can be obtained using the structured distance approach, but in general the condition of geometric uniformity is not a pre-requisite for the structured distance approach. Hence the codes obtained need not be GU codes.

As the code words are not selected to satisfy any specific property, the set of code words might not satisfy the rectangularity property.

Hence the codes obtained need not be rectangular. □

3.8 The Block Encoder

The structured distance approach gives a technique for obtaining general codes. Once a code suitable for a specific application is found, it has to be used in the communication system for that application. The block encoder transforms the data word to code words, using the block code of the BCM scheme. The code words are the channel symbols which are transmitted over the channel. The soft decoder is situated at the receiver and decodes the received

channel symbols to obtain the data words. This section discusses the implementation of block encoders for the general block codes of a BCM scheme obtained by using the structured distance approach. The soft decoding is discussed in Chapters 5 and 6.

3.8.1 Using a Code Table

Since the block codes are general and might not have any structure, one approach for block encoding can be to store the full code table in memory. The size of memory required will be directly proportional to the block length n and the number of code words $|C|$. The data words are used to address the memory and the code words will be stored at the addressed memory location. The symbols of the code words are mapped directly to the channel signals of the constellation provided by the modulator. This scheme is illustrated in Figure 3.8.1. For block codes of short length the simplicity of this scheme is its major advantage. But as the size of the code and the block length increases, the memory required for storing the look up table will become large.

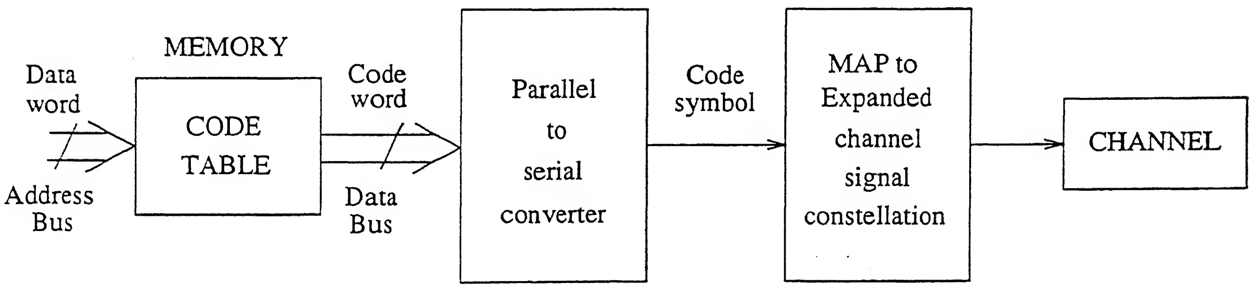

Figure 3.8.1: Block encoder using the code table

Example 3.8.1 Consider the block code obtained in Example 3.6.4.

The look up table for this will have to store 8, 4-valued 3-tuples. If binary-logic memory is used, the size of the memory required will be $8 \times 3 \times 2 = 48$ bits. The scheme is illustrated in Figure 3.8.2.

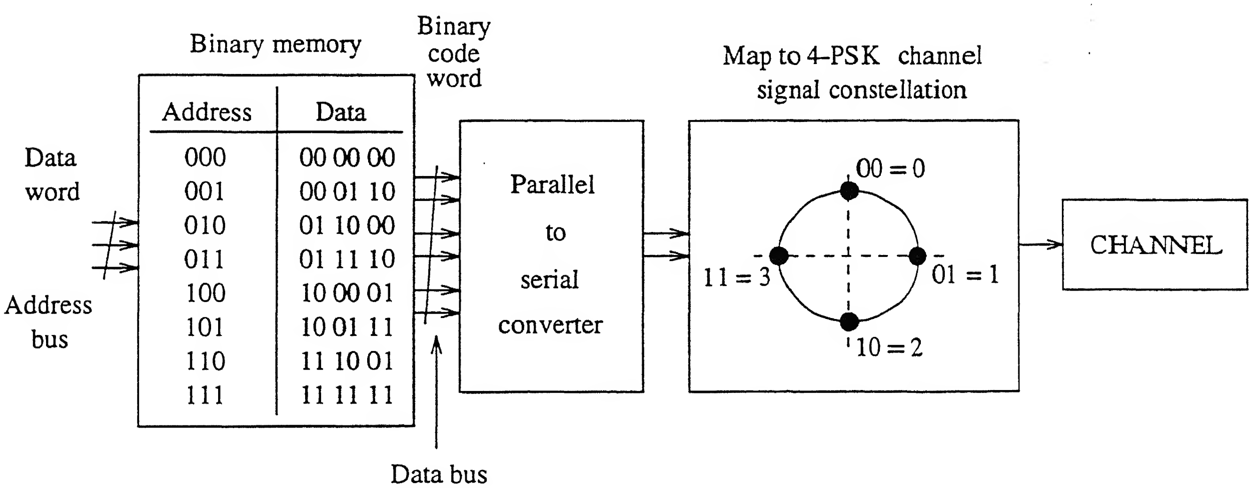

Figure 3.8.2: Block encoder using the code table for Example 3.8.1

3.8.2 Using Binary Combinational Logic

Another approach for the implementation of a block encoder, for the general codes obtained by the structured distance approach to BCM, can be the use of digital combinational logic. An intermediate binary representation for the data words and the code words can be employed. The combinational circuit maps from the binary data words to the binary code words. The binary code words are then mapped to the signals of the expanded channel signal constellation. This scheme is explained in the block diagram given in Figure 3.8.3. Very fast block encoders can be implemented using the combinational logic technique. Unlike the code table technique, the size of code or the block length does not constrains the use of this scheme. For large block length, large expanded channel signal constellation and large $|C|$, the complexity of the design of the combinational logic increases.

Example 3.8.2 Consider the block code obtained in Example 3.6.4.

Generally, the encoder can be directly designed from the code table, but here it is observed that a rearrangement of the code words can simplify the encoder design as illustrated in Table 3.8.1. In the table the mapping between the signals of the expanded channel signal constellation and the binary representation is as shown in Table 3.8.2.

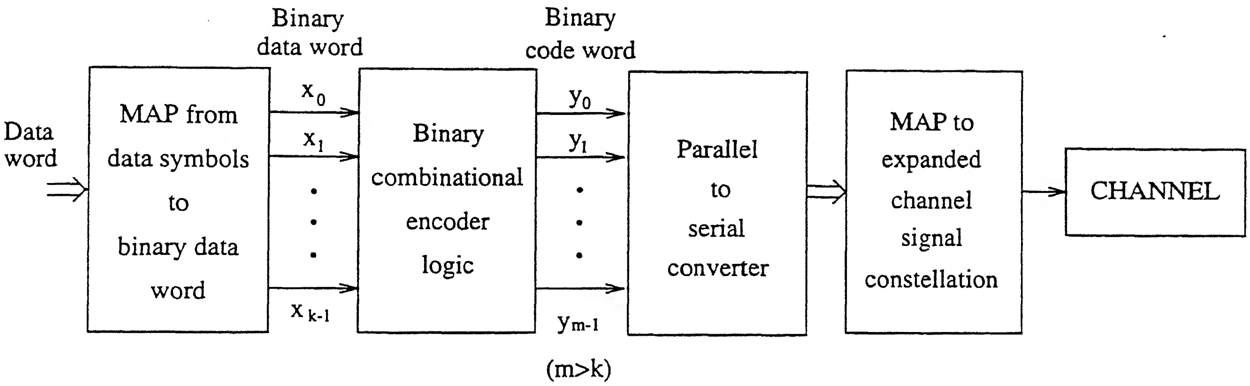

Figure 3.8.3: Block encoder using binary combinational logic

Table 3.8.1: Mapping from symbols to binary for the code in Example 3.8.2

Data word	Binary Data word ($x_2x_1x_0$)	Code word ($z_2z_1z_0$)	Binary code word ($y_5y_4y_3y_2y_1y_0$)
0	000	000	000000
1	001	012	000110
2	010	120	011000
3	011	132	011110
4	100	201	100001
5	101	213	100111
6	110	321	111001
7	111	333	111111

Table 3.8.2: Mapping from binary representation to channel symbols for code in Example 3.8.2

Binary representation	4-PSK Channel signal
00	0
01	1
10	2
11	3

From Table 3.8.1, the binary combinational block encoder circuit can be designed using the following equations.

$y_0 = x_2$,

$y_1 = x_0$,

$y_2 = x_0$,

$y_3 = x_1$,

$y_4 = x_1$ and

$y_5 = x_2$.

3.8.3 Considerations on the Design of the Block Encoder Based on Code Equivalence

Once a code has been found and the encoder for the code has been designed. The soft decoder for the block code has to be implemented. The issues related to the soft decoding of the general block codes are discussed at length in Chapters 5 and 6. A slight modification of the block encoder can simplify and result in a more efficient soft decoder. Hence, it is efficient to jointly design the encoder and decoder to optimize the BCM system. As discussed in Section 5.5.2, equivalent codes have different soft decoder structures, but are equivalent in the Euclidean distances distribution. So it is more efficient to use the equivalent code in the block encoder, as this speed up the soft decoding. Any block encoding scheme discussed in the previous sections can be employed with the equivalent code.

3.9 The Algorithm

This section summarizes the scheme developed under the structured distance approach.

(1) Define the BCM scheme required for the application by specifying,
   B' – the base signal constellation,
   S' – the expanded channel signal constellation,
   n – the block length of the code,
   $|C|$ – the number of code words and
   $d_{\min}$ – the minimum Euclidean distance for the code.

(2) Obtain $\hat{D}_{S' \times S' \times \ldots \times S' (n\text{-times})}$, the set of valid Euclidean distances for the distance distribution of the code.

(3) If the set $\hat{D}_{S' \times S' \times \ldots \times S' (n\text{-times})} = \emptyset$, then the scheme required in step (1) does not exists. Reselect a new BCM scheme by going back to step (1).

(4) Select distance n-tuples from the set $\hat{D}_{S' \times S' \times \ldots \times S' (n\text{-times})}$, which will be elements of the distance distribution of the code.

(5) Ensure that all the elements of distance distribution are valid and are elements of the set $\hat{D}_{S' \times S' \times \ldots \times S' (n\text{-times})}$. At this stage, if desired, some structure can be imposed on the distance distribution.

(6) Simultaneously assign signals, satisfying the distance distribution to obtain the code words.

(7) **IF** $|C|$ code words do not exist, **THEN** the scheme for such a distance distribution does not exists. Reselect a new distance distribution and repeat from stage (4) **OR** reselect a new BCM scheme by going back to step (1). **ELSE**

(8) Design a block encoder for the code.

(9) Obtain a reduced tree or minimal trellis for the code.[3]

(10) Design a soft maximum likelihood decoder for the code.

(11) Based on the considerations of the soft decoder redesign a block encoder for the equivalent code, if necessary.

3.10 Examples

The listings of various codes obtained by the structured distance approach are provided in the Appendix B. BCM schemes using a variety of channel signal constellations for various block lengths and $d_{\min}$ are presented in the appendix. A comparison between the various

[3] To be discussed in Chapters 5 and 6.

schemes based on the coding gain and the rate for the codes obtained is also tabulated. For the codes obtained all the possible values of $d_{\min}$ inside a proper range are considered.

For the purpose of illustration a few more example schemes are worked out in detail.

Example 3.10.1 Consider a BCM scheme with $n = 4$, $|C| = 16$ and required $d_{\min} \geq \sqrt{8}$ for the 4-PSK signal constellation given in Appendix A.

Code words starting with distance $d_0d_0d_2d_2$ and having $d_{\min} = 8$ can be obtained as follows, 0000, 0112, 2110, 1131, 1311, 1023, 1203, 3021, 3201, 0220. 2002, 0332, 2330, 3133, 3313, 2222.

The mapping from the data words to the code words is given in Table 3.10.1. The mapping

Table 3.10.1: Mapping from symbols to binary for the code in Example 3.10.1

Data word	Binary data word ($x_3x_2x_1x_0$)	Code word ($z_3z_2z_1z_0$)	Binary code word ($y_7y_6y_5y_4y_3y_2y_1y_0$)
0	0000	0000	00000000
1	0001	0112	00010110
2	0010	2110	10010100
3	0011	1131	01011101
4	0100	1311	01110101
5	0101	1023	01001011
6	0110	1203	01100011
7	0111	3021	11001001
8	1000	3201	11100001
9	1001	0220	00101000
10	1010	2002	10000010
11	1011	0332	00111110
12	1100	2330	10111100
13	1101	3133	11011111
14	1110	3313	11110111
15	1111	2222	10101010

between the signals of the expanded channel signal constellation and the binary representation is as shown in Table 3.10.2. The block encoder is obtained from the following equations,

$y_0 = \bar{x}_3x_2 + x_2\bar{x}_1x_0 + \bar{x}_3x_1x_0 + x_2x_1\bar{x}_0 + x_3\bar{x}_2\bar{x}_1\bar{x}_0,$

Table 3.10.2: Mapping from binary representation to channel symbols for the code in Example 3.10.1

Binary representation	4-PSK Channel signal
00	0
01	1
10	2
11	3

$y_1 = x_3x_1 + x_3x_2x_0 + \bar{x}_3\bar{x}_1x_0 + x_2x_1\bar{x}_0,$

$y_2 = \bar{x}_3\bar{x}_2x_0 + \bar{x}_3\bar{x}_2x_1 + \bar{x}_2x_1x_0 + x_2\bar{x}_1\bar{x}_0 + x_3x_2\bar{x}_0 + x_3x_2\bar{x}_1,$

$y_3 = x_1x_0 + x_3x_0 + x_2x_0 + x_3x_2\bar{x}_1,$

$y_4 = y_2,$

$y_5 = x_3\bar{x}_1\bar{x}_0 + x_3\bar{x}_2\bar{x}_1 + x_3x_1x_0 + x_3x_2x_1 + \bar{x}_3x_2\bar{x}_0,$

$y_6 = y_0$ and

$y_7 = x_3x_2 + x_3\bar{x}_0 + \bar{x}_2x_1\bar{x}_0 + x_2x_1x_0.$

Example 3.10.2 Consider that a block code with $n = 3$, $|C| = 7$ and a required $d_{\text{min}} \geq \sqrt{6}$ for the 5-PSK signal constellation shown in Appendix A.

For the channel signal constellation the Euclidean distances are $d_0 = 0$, $d_1 = \sqrt{1.38}$ and $d_2 = \sqrt{3.62}$. The code words with a $d_{\text{min}} = \sqrt{6.38}$ obtained are as follows,
000, 112, 443, 314, 241, 421, 134.
The mapping from the data words to the code words is given in Table 3.10.3. The mapping between the signals of the expanded channel signal constellation and the binary representation is as shown in Table 3.10.4. The block encoder is obtained from the following equations,

$y_0 = x_2\bar{x}_1 + \bar{x}_2x_1\bar{x}_0,$

$y_1 = \bar{x}_2\bar{x}_1x_0 + \bar{x}_2x_1\bar{x}_0,$

$y_2 = x_1x_0 + x_2x_1,$

$y_3 = \bar{x}_2x_0 + x_2x_1,$

$y_4 = x_2x_0 + x_2x_1,$

$y_5 = \bar{x}_2x_1\bar{x}_0 + x_2\bar{x}_1\bar{x}_0,$

$y_6 = y_3,$

Table 3.10.3: Mapping from symbols to binary for the code in Example 3.10.2

Data word	Binary data word ($x_2x_1x_0$)	Code word ($z_2z_1z_0$)	Binary code word ($y_8y_7y_6y_5y_4y_3y_2y_1y_0$)
0	000	000	000000000
1	001	112	001001010
2	010	443	100100011
3	011	314	011001100
4	100	241	010100001
5	101	421	100010001
6	110	134	001011100

Table 3.10.4: Mapping from binary representation to channel symbols for the code in Example 3.10.2

Binary representation	5-PSK Channel signal
000	0
001	1
010	2
011	3
100	4

$y_7 = x_1x_0 + x_2\bar{x}_1\bar{x}_0$, and

$y_8 = x_2x_0 + \bar{x}_2x_1\bar{x}_0$.

3.11 Concluding Remarks

A new viewpoint termed as the structured distance approach is proposed for obtaining codes for BCM schemes. The technique developed is general and can be used with arbitrary channel signal constellation based BCM schemes. The same scheme can work with any channel signal constellation. The scheme is also general as it results in general codes which may or may not be linear codes, cyclic codes, group codes, GU codes or rectangular codes.

The main advantage gained by the generality is that a wide range of codes can be obtained.[4] The main disadvantage of the scheme is that the code table is required for encoding and decoding. A simple scheme for the block encoder and an algorithm for the scheme have been obtained. Codes have been reported with redundancy in space and time and with various signal constellations which include asymmetric signal constellations and constellations with number of signals not a prime or a power of prime.

[4]A few are listed in Appendix B.

Chapter 4

Sphere Packings and Block Coded Modulation

4.1 Introduction

This chapter proposes a new class of codes based on the structured distance approach. These codes known as **codes based on the selective permutations of distances**, use an arbitrary expanded channel signal constellation.

The structured distance approach, is used, with some results from sphere packings to obtain block codes for BCM schemes. A structure based on results from sphere packings is enforced on the Euclidean distances to obtain the distance distribution of the code.

To begin with, a brief overview of some essential terminology related with sphere packings is introduced. The important results, concerning sphere packings, are summarized in Appendix C. The analogy between the coding problem and sphere packings is used for stating some properties of codes for BCM schemes and for influencing the search for codes. A condition on distances for the distance distribution of a code is arrived at from considerations of sphere packings. The structure of the codes is some what similar to the structure of spherical codes [1], only now, the codes are considered using the Euclidean distance metric instead of the Hamming distance metric.

The class of block codes proposed in this chapter, results in general codes for arbitrary channel signal constellations. The technique is summarized in an algorithm and some illustrative examples are given. The chapter ends with a few concluding remarks. Soft decoding

of the codes obtained in this chapter is dealt with in Chapters 5 and 6.

4.2 Background and Preliminaries

The problems of coding and sphere packing have a lot in common. It has been known since the work of Nyquist and Shannon that the design of optimal codes for band-limited AWGN channel is equivalent to the sphere packing problem. For conventional codes based on the Hamming distance metric, a lot of work relating it to sphere packings has been done, which is summarized in the book [21]. Not much work exists on sphere packings used with codes based on the Euclidean distance metric.

The problem of finding codes for BCM, as in the case of error-correcting codes, is analogous to the sphere packing problem [21,55,77]. Results from packing of equal spheres [21,68], are used for TCM [20, 25, 26] to obtain signal constellations which are dense.

The sphere packing problem[1], consists of arranging equal spheres in the $\tilde{n}$-dimensional space, such that no two spheres of the system have any inner point in common.

Definition 4.2.1 *A problem of obtaining codes for a BCM scheme for AWGN channels can be termed as, finding some finite number of code words (points) in some finite Euclidean discrete space, such that the minimum Euclidean distances $d_{\min}$ between the code words (points) is maximized.*

The following points can be noted regarding the analogy between the sphere packing problem and the problem of finding codes.

- Coding ensures that the distance between code words is $\geq d_{\min}$. So with each code word if a sphere is associated with the center as the code word and the radius of the sphere as $d_{\min}/2$, then these equal spheres for a code cannot intersect.
- The number of spheres equals the number of code words.
- The space in which the code words exist is some $\tilde{n}$-dimensional Euclidean space, where $\tilde{n}$ depends on the dimensionality of the expanded channel signal constellation and the

[1]Defined in Appendix C.

block length of the code. Note that, the matrix of Euclidean distances between the sequences of finite signals from a finite signal constellation $d_{S\times S\times\ldots\times S(n\text{-times})}$, discussed in Section 2.3 defines this space.

- The Euclidean space of the code words is finite, as for a BCM scheme the channel is assumed to be power-limited. This limit on the signal power bounds the Euclidean space of the code words. Since the same signal constellation is used in all the n-dimensions, this bounded region will be a $\tilde{n}$-dimensional cube. Note that, this basically means that the elements in each row of the matrix $\mathbf{d_S}$ defined in Section 2.2 is bounded to a value depending on the power constraint.

- The Euclidean space of the code words is discrete, as only the points provided by the expanded channel signal constellation in the $\tilde{n}$-dimensional space can be the centers of the code words. This is a consequence of the assumption that the channel is discrete input, analog output. The received words at the receiver of the channel are in the continuous bounded Euclidean space.

- Maximization of the minimum Euclidean distance implies maximization of the radius of the spheres forming the packing.

The space between the spheres, depending on the proximity of the points in it with the center, is assigned to a particular center. The regions thus formed are convex and are known as Voronoi regions[2] [2, 21].

Definition 4.2.2 *The space (volume) of a Voronoi region, outside the sphere associated with the code word, is known as the interstitial space for a code word.*

The Voronoi regions can be characterized by the following statements.

- One and only one sphere of radius $d_{\min}/2$ associated with a code word is inscribed in a Voronoi region.

- No other sphere, centered at any other code word, can intersect the Voronoi region of a code word.

[2]The Voronoi regions are also known as the maximum likelihood regions or soft-decoding regions associated with a code word.

- For a block code, denser the packing, lesser is the interstitial space, which does not contribute to $d_{\min}$, but more will be the nearest neighbor for a code word.

4.3 Sphere Packings and BCM

Definition 4.3.1 *For the uncoded case, the spheres with all the data words as center, lie in a $\tilde{n}_b$-dimensional discrete finite Euclidean space, where $\tilde{n}_b$ depends on the dimensionality of the base signal constellation and the block length of the data words. The spheres with radius $d_{uc}/2$ form a packing.*

Definition 4.3.2 *The code words lie in a $\tilde{n}$-dimensional discrete finite Euclidean space, where $\tilde{n}$ depends on the dimensionality of the expanded channel signal constellation and the block length n of the code.*

Theorem 4.3.1 $\tilde{n} \geq \tilde{n}_b$.

Proof: For any code, the block length $n \geq$ block length for data words.
From Definition 3.3.2, for an expanded channel signal constellation $n' \geq n'_b$.
This implies that the dimensionality of the channel signal constellation $\geq$ the dimensionality of the base signal constellation.
Hence, $\tilde{n} \geq \tilde{n}_b$. □

Definition 4.3.3 *Let the number of points in the $\tilde{n}_b$-dimensional discrete finite Euclidean space of the data words be N_b.*

Definition 4.3.4 *The code words lie in a $\tilde{n}$-dimensional discrete finite Euclidean space, consisting of N_c points.*

Theorem 4.3.2 $N_c \geq N_b$.

Proof: For any code, the block length $n \geq$ block length for data words.
From Definition 3.3.2, for an expanded channel signal constellation $n' \geq n'_b$.
Also from Theorem 4.3.1, $\tilde{n} \geq \tilde{n}_b$.
Hence, $N_c \geq N_b$. □

Example 4.3.1 *Consider the (2-PSK, 4-PSK, 3, 8, $\sqrt{6}$) code obtained in Example 3.6.4.*

For this code,
The block length of the data words = The block length of the code words; $n = 3$.
The number of signals in the 2-PSK base signal constellation, $n'_b = 2$, and the number of signals in the 4-PSK expanded channel signal constellation, $n' = 4$.
The base signal constellation is 1-dimensional and the expanded channel signal constellation is 2-dimensional.
The $\tilde{n}_b$ = 3-dimensional discrete Euclidean space of the data words contains $N_b = {n'_b}^n = 2^3$ = 8 points, that is data words.
The $\tilde{n}$ = 6-dimensional discrete Euclidean space of the code words contains $n'^n = 4^3 = 64$ points.

Consider the following cases of BCM schemes.

(1) If n increases, both $\tilde{n}_b$ and $\tilde{n}$ increase, and hence, the dimensionality of the discrete Euclidean space of the data words, the number of data words N_b and the dimensionality of the discrete Euclidean space of the code words increase. If the number of data words is reduced and only a subset of the data words is considered ($< N_b$), then this case corresponds to **redundancy in time.**

(2) As n', the number of signals in the expanded channel signal constellation is more than the number of signals in the base signal constellation n'_b, hence, the number of points in the discrete Euclidean space of the code words N_c are more than the number of data words N_b. If of the N_c points N_b are selected to obtain a $d_{\min} > d_{\text{uc}}$, then this case corresponds to **redundancy in space.**

(3) As n', the number of signals in the expanded channel signal constellation is more than the number of signals in the base signal constellation n'_b, hence, the number of points in the discrete Euclidean space of the code words N_c are more than the number of data words N_b. If of the N_c points, points less than N_b are selected to obtain a $d_{\min} > d_{\text{uc}}$, then this case corresponds to **redundancy in space and time.**

(4) For the efficient utilization of the redundancy in the Euclidean space, it is better if the dimensionality of the space increases without an increase in the number of data

words N_b. This is possible when the expansion of the signal set. to obtain the expanded channel signal constellation from the base signal constellation, results in an expansion of the dimensionality of the Euclidean space. For example, if the base signal constellation is 2-PSK and the expanded channel signal constellation is 4-PSK, then the expansion of the signal constellation results in the increase in dimensionality of $\tilde{n}$. This results in more space (as the dimension increases), with increasing the number of data words, as compared to using the 4-PSK as the base signal constellation and 8-PSK as the expanded channel signal constellation. Hence, **multi-dimensional channel signal constellations** are better suited for BCM schemes.

Finding codes for a BCM scheme consists of arranging $|C|$ spheres with radius $d_{\text{min}}/2$, where $d_{\text{min}} \geq d_{\text{uc}}$, in the $\tilde{n}$-dimensional discrete Euclidean space consisting of N_c points, so that the spheres form a packing.

The number of spheres forming the packing and the radius of the spheres are the parameters of interest. The coding gain of Definition 3.3.7 is a consequence of the increase in d_{min}, or the radius of the spheres of the packing. Depending on the number of spheres of the packing the rate of the coding scheme follows.

(1) For rate, $R = 1$, the number of spheres in the packing corresponding to the code is equal to the number of the data words. Packing results in a rearrangement of the spheres in the $\tilde{n}$-dimensional discrete Euclidean space of N_c points, to increase the radius of the spheres to $d_{\text{min}}/2$.

(2) For rate, $R < 1$, the number of spheres in the packing corresponding to the code is less than the number of the data words. Packing results in an arrangement of the spheres in the $\tilde{n}$-dimensional discrete Euclidean space of N_c points, to increase the radius of the spheres to $d_{\text{min}}/2$.

(3) For rate, $R > 1$, the number of spheres in the packing corresponding to the code is more than the number of the data words. Packing results in an arrangement of the spheres in the $\tilde{n}$-dimensional discrete Euclidean space of N_c points, to increase the radius of the spheres to $d_{\text{min}}/2$.

Consider the following points from the packing of spheres.

- The best arrangement of the spheres might not be a lattice [21].
- It cannot be inferred that, a packing which increases the contact number is the most dense packing [55].
- In general, the ideal situation might be, the use of the continuous finite Euclidean space R^n, to obtain the best packing of spheres. This corresponds to using analog signals and mapping data words to these signals.

These considerations from sphere packings are used to develop a scheme to obtain codes for BCM.

4.4 Obtaining Codes for BCM

Viewed in the frame work of sphere packing, the problem of obtaining block codes for BCM is analogous to obtaining a sphere packing. Once the block length n is specified, the base signal constellation, defines the $\tilde{n}_b$-dimensional discrete Euclidean space of the data words and a selection of $|C|$, determines the number of data words N_b. The expanded channel signal constellation specifies the dimensionality $\tilde{n}$ of the discrete Euclidean space of the code words having N_c points. The required $d_{\min}$ fixes the radius of the spheres, which forms a packing denoting the code. The centers of the spheres will be the code words. In this way a sphere packing can be found out to obtain a code for a BCM scheme.

In order to obtain a sphere packing corresponding to the coding problem specified by the application, though a packing which increases the contact number might not be the most dense packing in general, for the specific case of the bounded discrete Euclidean space of a BCM scheme, attention is restricted to packings, which increase the contact number in the finite volume.

Definition 4.4.1 *A sphere with center at one of the N_c points of the $\tilde{n}$-dimensional discrete Euclidean space bounded by the $\tilde{n}$-dimensional cube*[3] *is known as a relevant sphere.*

[3]Due to power constraint.

All the spheres are of radius $d_{\min}/2$.
Let, $\tilde{C} = \{\tilde{C}_1, \tilde{C}_2, \cdots, \tilde{C}_{n'^n}\}$ be the set of relevant spheres, where the cardinality of the set $\tilde{C}$ is $n'^n \geq |C|$.
n' = The number of signals in the expanded channel signal constellation.
n = The block length of the code.
$|C|$ = The number of code words.
Let, c_i = The center of the i^{th} sphere $\tilde{C}_i$.
$S' \times S' \times \ldots \times S'(n\text{-times})$ = The set of sequences of the expanded channel signal constellation of length n, of cardinality n'^n.
Then, $c_i \in S' \times S' \times \ldots \times S'(n\text{-times})$.
Let, τ_i = The contact number of the sphere $\tilde{C}_i$ with center c_i.

Theorem 4.4.1 *For $|C|$ relevant spheres, the radius of the sphere $d_{\min}/2$ is maximized, when $\sum_{i=1}^{|C|} \tau_i$ is maximized.*

Proof: The volume of a $\tilde{n}$-dimensional sphere of radius $d_{\min}/2$ is

$$V = V_{\tilde{n}} \left(\frac{d_{\min}}{2} \right)^{\tilde{n}}, \quad \text{where,} \quad V_{\tilde{n}} = \frac{\pi^{\frac{1}{2}\tilde{n}}}{\Gamma\left(\frac{1}{2}\tilde{n} + 1\right)}.$$

Therefore, maximization of $d_{\min}/2 \quad \Rightarrow \quad$ maximization of the volume of the sphere.
Let, V_{S_i} be the volume of the i^{th} sphere, lying inside the $\tilde{n}$-dimensional cube, $1 \leq i \leq |C|$.
Let, V_i denote the volume of the i^{th} Voronoi region inside the $\tilde{n}$-dimensional cube, $1 \leq i \leq |C|$.
The i^{th} Voronoi region is a convex region in the $\tilde{n}$-dimensional Euclidean space, partitioning the volume of the $\tilde{n}$-dimensional cube. The i^{th} Voronoi region contains the points of the Euclidean space which are closer to the center c_i than to any other center. These regions are made up of hyper-planes, and any region will have at most $n'^n - 1$ hyper-planes.
Now, $\sum_{i=1}^{|C|} V_i = k$, the total volume of the cube (a constant).
For the i^{th} Voronoi region, the interstitial space is $(V_i - V_{S_i})$.
Therefore, $\sum_{i=1}^{|C|}(V_i - V_{S_i}) + \sum_{i=1}^{|C|} V_{S_i} = k$

$$\text{Hence, maximization of } d_{\min}/2 \quad \Rightarrow \quad \text{minimization of } \sum_{i=1}^{|C|}(V_i - V_{S_i}). \tag{4.4.1}$$

The sphere i is inscribed inside the Voronoi region i, such that c_i is the center of the sphere. This implies that $(V_i - V_{S_i})$ is minimum, when the sphere touches maximum hyper-planes of the Voronoi region i.
Let this number be T_i.
Now, $T_i \geq \tau_i \quad 1 \leq i \leq |C|$.
Hence to minimize 4.4.1, maximize $\sum_{i=1}^{|C|} \tau_i$. □

This result can be used to obtain codes, from sphere packings. An arrangement is required which maximizes the sum of contact numbers of the spheres. The problem of obtaining such arrangement of spheres is the primary motivation for the following section. The result can be used as the condition for structuring the Euclidean distances to obtain the distance distribution of a code. So a new class of codes belonging to the general codes of the structured distance approach, can be found.

4.5 Codes Based on the Selective Permutations of Distances

This section uses Theorem 4.4.1 with the structured distance approach proposed in Chapter 3, to obtain a class of codes to be used with BCM schemes. The objective is to obtain sphere packings which increase the contact number in the finite discrete Euclidean space. The primary motivation is the observation of the following fact.

- All the permutations of a distance n-tuple, $d_{i_1}d_{i_2}\ldots d_{i_n}$ give,

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} = d_I.$$

Definition 4.5.1 *If $d_{i_1}d_{i_2}\ldots d_{i_n}$ is an n-tuple, then the set consisting of all the permutations of $d_{i_1}d_{i_2}\ldots d_{i_n}$ is $\tilde{S}_{d_{i_1}d_{i_2}\ldots d_{i_n}}$.*

Theorem 4.5.1 *If a distance n-tuple $d_{i_1}d_{i_2}\ldots d_{i_n}$ is such that*

$$d_{i_1}d_{i_2}\ldots d_{i_n} \in \hat{D}_{S'\times S'\times\ldots\times S'\,(n\text{-times})},$$

then the set of valid distances,

$$\tilde{S}_{d_{i_1}d_{i_2}\ldots d_{i_n}} \subset \hat{D}_{S'\times S'\times\ldots\times S'\,(n\text{-times})}.$$

Proof:

$$d_{i_1} d_{i_2} \dots d_{i_n} \in \hat{D}_{S' \times S' \times \dots \times S'(n\text{-times})}$$

implies that

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \dots + d_{i_n}^2} \geq d_{\min}.$$

Also, all the permutations result in,

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \dots + d_{i_n}^2} \geq d_{\min}.$$

Hence, all the elements of the set $\tilde{S}_{d_{i_1} d_{i_2} \dots d_{i_n}}$ have

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \dots + d_{i_n}^2} \geq d_{\min}.$$

Hence,

$$\tilde{S}_{d_{i_1} d_{i_2} \dots d_{i_n}} \subset \hat{D}_{S' \times S' \times \dots \times S'(n\text{-times})}.$$

□

Lemma 4.5.1 *If $d_{i_1} d_{i_2} \dots d_{i_n}$ is valid and,*

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \dots + d_{i_n}^2} = d_{\min},$$

then all the permutations of $d_{i_1} d_{i_2} \dots d_{i_n}$ are valid and result in,

$$\sqrt{d_{i_1}^2 + d_{i_2}^2 + \dots + d_{i_n}^2} = d_{\min}.$$

As a consequence of this lemma, all the permutations can be considered to give spheres such that the spheres touch a central sphere. In fact, considering all permutations of the distance n-tuple is same as restricting to symmetric arrangements of spheres around a sphere. So, all permutations of the distance n-tuples can be used to obtain spherical codes with the Euclidean distance metric.

Example 4.5.1 *For example consider the 4-PSK signal constellation shown in Appendix A.*

Consider a block length $n = 3$ and $d_{\min} = \sqrt{6}$.
The Euclidean distances between the signals of the 4-PSK signal constellation are denoted

by $d_0 = 0$, $d_1 = \sqrt{2}$ and $d_2 = 2$.
The set of Euclidean distances is $D = \{ d_0, d_1, d_2 \}$.
The set of all the 3-tuples of distances is,
$D_{S'\times S'\times S'} = \{ d_0d_0d_0, d_0d_0d_1, d_0d_0d_2, d_0d_1d_0, d_0d_1d_1, d_0d_1d_2, d_0d_2d_0, d_0d_2d_1, d_0d_2d_2, d_1d_0d_0, d_1d_0d_1, d_1d_0d_2, d_1d_1d_0, d_1d_1d_1, d_1d_1d_2, d_1d_2d_0, d_1d_2d_1, d_1d_2d_2, d_2d_0d_0, d_2d_0d_1, d_2d_0d_2, d_2d_1d_0, d_2d_1d_1, d_2d_1d_2, d_2d_2d_0, d_2d_2d_1, d_2d_2d_2 \}$.
The set of valid Euclidean distances is,
$\hat{D}_{S'\times S'\times S'} = \{ d_0d_1d_2, d_0d_2d_1, d_0d_2d_2, d_1d_0d_2, d_1d_1d_1, d_1d_1d_2, d_1d_2d_0, d_1d_2d_1, d_1d_2d_2, d_2d_0d_1, d_2d_0d_2, d_2d_1d_0, d_2d_1d_1, d_2d_1d_2, d_2d_2d_0, d_2d_2d_1, d_2d_2d_2 \}$.
Each element of $\hat{D}_{S'\times S'\times S'}$ is a Euclidean distance $\geq d_{\min}$.
Now consider the 3-tuple, $d_0d_1d_2 \in \hat{D}_{S'\times S'\times S'}$.
The set of all the permutations of $d_0d_1d_2$ is,
$\tilde{S}_{d_0d_1d_2} = \{ d_0d_1d_2, d_0d_2d_1, d_1d_0d_2, d_1d_2d_0, d_2d_0d_1, d_2d_1d_0 \}$.
And. $\tilde{S}_{d_0d_1d_2} \subset \hat{D}_{S'\times S'\times S'}$. Also,

$$\begin{aligned}
\sqrt{d_0^2 + d_1^2 + d_2^2} &= \sqrt{d_0^2 + d_2^2 + d_1^2} \\
&= \sqrt{d_1^2 + d_0^2 + d_2^2} \\
&= \sqrt{d_1^2 + d_2^2 + d_0^2} \\
&= \sqrt{d_2^2 + d_0^2 + d_1^2} \\
&= \sqrt{d_2^2 + d_1^2 + d_0^2} \\
&= d_{\min}.
\end{aligned}$$

Definition 4.5.2 *If a distance n-tuple $d_{X_1y_{i_1}}d_{X_2y_{i_2}}\ldots d_{X_ny_{i_n}}$ is such that,*

$$\sqrt{d^2_{X_1y_{i_1}} + d^2_{X_2y_{i_2}} + \ldots + d^2_{X_ny_{i_n}}} = d_{\min}.$$

The spheres with centers as the code words $X_1X_2\ldots X_n$ and $y_{i_1}y_{i_2}\ldots y_{i_n}$ are of radius $d_{\min}/2$, and touch each other.

Theorem 4.5.2 *All the permutations of the distance n-tuple $d_{X_1y_{i_1}}d_{X_2y_{i_2}}\ldots d_{X_ny_{i_n}}$ result in spheres which are obtained by rotations of the sphere with center at $y_{i_1}y_{i_2}\ldots y_{i_n}$ around the sphere with center at $X_1X_2\ldots X_n$ in the n-dimensional Euclidean space.*

Proof: Any permutation of the distance n-tuple, $d_{X_1y_{i_1}}d_{X_2y_{i_2}}\ldots d_{X_ny_{i_n}}$ with the point $X_1X_2\ldots X_n$ fixed, results in the transformation of a point

$$y_{i_1}y_{i_2}\ldots y_{i_n} = Y_{1_1}Y_{1_2}\ldots Y_{1_n}$$

to another point

$$y_{i_1}y_{i_2}\ldots y_{i_n} = Y_{2_1}Y_{2_2}\ldots Y_{2_n}$$

such that,

$$\begin{aligned}\sqrt{d^2_{X_1y_{i_1}}+d^2_{X_2y_{i_2}}+\ldots+d^2_{X_ny_{i_n}}} &= \sqrt{d^2_{X_1Y_{1_1}}+d^2_{X_2Y_{1_2}}+\ldots+d^2_{X_nY_{1_n}}}\\ &= \sqrt{d^2_{X_1Y_{2_1}}+d^2_{X_2Y_{2_2}}+\ldots+d^2_{X_nY_{2_n}}}\\ &= d_{\min}.\end{aligned}$$

Hence, all the permutations of the distance n-tuple $d_{X_1y_{i_1}}d_{X_2y_{i_2}}\ldots d_{X_ny_{i_n}}$ results in spheres, which are obtained by rotations of the sphere with center at $y_{i_1}y_{i_2}\ldots y_{i_n}$ around the sphere with center at $X_1X_2\ldots X_n$, in the n-dimensional Euclidean space. □

As a consequence of this, a rotationally symmetric arrangement of spheres, which increases the contact number with a central sphere is obtained by considering all the permutations of a valid distance n-tuple.

Theorem 4.5.3 *The arrangement of spheres obtained by considering all the permutations of the distance n-tuple $d_{X_1y_{i_1}}d_{X_2y_{i_2}}\ldots d_{X_ny_{i_n}}$, which results in spheres, obtained by rotations of the sphere with center at $y_{i_1}y_{i_2}\ldots y_{i_n}$ around the sphere with center at $X_1X_2\ldots X_n$, in the n-dimensional Euclidean space, need not form a packing.*

Proof: Any permutation of the distance n-tuple, $d_{X_1y_{i_1}}d_{X_2y_{i_2}}\ldots d_{X_ny_{i_n}}$ with the point $X_1X_2\ldots X_n$ fixed, results in the transformation of a point

$$y_{i_1}y_{i_2}\ldots y_{i_n} = Y_{1_1}Y_{1_2}\ldots Y_{1_n}$$

to another point

$$y_{i_1}y_{i_2}\ldots y_{i_n} = Y_{2_1}Y_{2_2}\ldots Y_{2_n}$$

such that,

$$\begin{aligned}\sqrt{d^2_{X_1y_{i_1}}+d^2_{X_2y_{i_2}}+\ldots+d^2_{X_ny_{i_n}}} &= \sqrt{d^2_{X_1Y_{1_1}}+d^2_{X_2Y_{1_2}}+\ldots+d^2_{X_nY_{1_n}}}\\ &= \sqrt{d^2_{X_1Y_{2_1}}+d^2_{X_2Y_{2_2}}+\ldots+d^2_{X_nY_{2_n}}}\\ &= d_{\min}.\end{aligned}$$

But the spheres at center $Y_{1_1}Y_{1_2}\ldots Y_{1_n}$ and $Y_{2_1}Y_{2_2}\ldots Y_{2_n}$ can intersect.
That is, the composition of a valid distance n-tuple and its permutation, need not be valid. Hence, the arrangement of spheres obtained by considering all the permutations of a distance n-tuple, need not form a packing. □

Example 4.5.2 *Consider the discussion of Example 4.5.1 for the 4-PSK signal constellation.*

For $d_{\min} = \sqrt{6}$, the set of valid Euclidean distances is,
$\hat{D}_{S'\times S'\times S'} = \{\, d_0d_1d_2,\ d_0d_2d_1,\ d_0d_2d_2,\ d_1d_0d_2,\ d_1d_1d_1,\ d_1d_1d_2,\ d_1d_2d_0,\ d_1d_2d_1,\ d_1d_2d_2,\ d_2d_0d_1,\ d_2d_0d_2,\ d_2d_1d_0,\ d_2d_1d_1,\ d_2d_1d_2,\ d_2d_2d_0,\ d_2d_2d_1,\ d_2d_2d_2\,\}$.
Each element of $\hat{D}_{S'\times S'\times S'}$ is a Euclidean distance $\geq d_{\min} = \sqrt{6}$.
The set of all the permutations of $d_0d_1d_2$ is,
$\tilde{S}_{d_0d_1d_2} = \{\, d_0d_1d_2,\ d_0d_2d_1,\ d_1d_0d_2,\ d_1d_2d_0,\ d_2d_0d_1,\ d_2d_1d_0\,\}$.
And, $\tilde{S}_{d_0d_1d_2} \subset \hat{D}_{S'\times S'\times S'}$.
But, for $d_0d_1d_2 \in \tilde{S}_{d_0d_1d_2}$ and $d_0d_2d_1 \in \tilde{S}_{d_0d_1d_2}$,
$d_0d_1d_2 \circ d_0d_2d_1 = d_0d_1d_1 \notin \hat{D}_{S'\times S'\times S'}$.

As a consequence of Theorem 4.5.3, it is not possible to use all the permutations of a distance n-tuple, but to select only those permutations which results in a sphere packing. The distance distribution of a code is obtained by selecting these permutations of valid distances. So the class of codes obtained is known as **codes based on selective permutations of distances.** Therefore, in order to obtain block codes for the BCM scheme, specified by the application, attention is restricted to rotational symmetrical arrangements of spheres forming a packing. In this manner, selective permutations of distances as opposed to all permutations of signals discussed by Slepian [75], are used in the construction of the distance distribution, and the code.

To obtain code words, for a specified expanded channel signal constellation, block length n, and $d_{\min}$, first the set of valid distance n-tuples for the distribution of the code is found. An initial code word is first assumed. Choosing a valid distance, other distances are obtained by using all permutations of the n-tuple corresponding to the distance, taking care so that all the distances of the distribution remain valid. To obtain maximum code words at each stage of the search, it may be necessary to obtain the full set of code words including the

invalid ones, and then make a proper choice, resulting in a larger number of valid code words. If instead of only maximizing the minimum distance, it is also of interest to minimize the number of nearest neighbors, then the other distances larger than $d_{\min}$, can be used for permutations. This scheme based on the selective permutations of distances of a constellation can be used with an arbitrary channel signal constellation. The scheme is summarized in an algorithm in the following section.

4.6 The Algorithm

The scheme for obtaining the class of codes, based on the selective permutations of distances, using the structured distance approach can be summarized in the following algorithm.

(1) Define the BCM scheme required for the application by specifying,
B' – the base signal constellation,
S' – the expanded channel signal constellation,
n – the block length of the code,
$|C|$ – the number of code words and
$d_{\min}$ – the minimum Euclidean distance for the code.

(2) Obtain $\hat{D}_{S'\times S'\times\ldots\times S'(n\text{-times})}$, the set of valid Euclidean distances for the distance distribution of the code.

(3) If the set $\hat{D}_{S'\times S'\times\ldots\times S'(n\text{-times})} = \emptyset$, then the scheme required in step (1) does not exists. Reselect a new BCM scheme by going back to step (1).

(4) Select a distance n-tuple, $d_{i_1}d_{i_2}\ldots d_{i_n}$ from the set $\hat{D}_{S'\times S'\times\ldots\times S'(n\text{-times})}$, which will be element of the distance distribution of the code. Depending on the requirement of the code[4] this distance n-tuple will be such that, $\sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} = d_{\min}$.

(5) Select an initial code word and simultaneously obtain other code words satisfying the distance distribution.

[4]Another requirement for the code, can be to minimize the nearest neighbors of a code, then, this element will have to be chosen such that $\sqrt{d_{i_1}^2 + d_{i_2}^2 + \ldots + d_{i_n}^2} > d_{\min}$.

(6) Obtain $\tilde{S}_{d_{i_1} d_{i_2} \ldots d_{i_n}}$, the set of all permutations of the selected distance n-tuple.

(7) Select elements from $\tilde{S}_{d_{i_1} d_{i_2} \ldots d_{i_n}}$, such that, all the elements of distance distribution are valid and are elements of the set $\hat{D}_{S' \times S' \times \ldots \times S' (n\text{-times})}$.

(8) If after using all the selective permutations of the chosen distance n-tuples, $|C|$ code words are not found, select another distance n-tuple, and with this again obtain more distances and code words by going back to step (4). **REPEAT** this till $|C|$ code words are found.

(9) If $|C|$ code words do not exist under this scheme, then reselect a new BCM scheme by going back to step (1).

(10) Design a block encoder for the code as explained in Section 3.8.

(11) Obtain a reduced tree or minimal trellis for the code.[5]

(12) Design a soft maximum likelihood decoder for the code.

This algorithm searches for a set of code words for a BCM scheme, such that the spheres associated with the code words form a packing having rotational symmetry and the sum of the contact number of the valid spheres is maximized.

4.7 Properties of the Codes

The codes based on the selective permutations of distances forms a class of the general codes obtained by the structured distance approach. A few important properties of this class of codes are now specified.

Theorem 4.7.1 *The codes based on the selective permutations of distances need not be linear, cyclic, lattice, group, GU or rectangular codes.*

Proof: In the structured distance approach, using the selective permutations of distances the codes are obtained from Euclidean distances.

[5]To be discussed in Chapters 5 and 6.

As the code words are not selected to satisfy any specific property, there might not be any linear transformation between the data words and the code words. Hence the codes obtained need not be linear. Also the codes need not be cyclic.
The rotationally symmetric arrangements of sphere packing can be such that it does not forms a lattice. Hence the codes obtained need not be lattice codes.
As the code words are not selected to satisfy any specific property, the set of code words might not form a sub-group. Hence the code obtained need not be a group code.
GU codes are codes for which the Euclidean distances are such that, the Voronoi regions [21] for all the code words are identical, and separated by just rotations and translations in the Euclidean space. A distance distribution can be selected such that this condition is satisfied. So a GU code can be obtained using the structured distance approach with the selective permutations of distances, but in general the condition of geometric uniformity is not a pre-requisite for the structured distance approach. Also selective permutations of distances can be used with an expanded channel signal constellation, which might not be GU. Hence the codes obtained need not be GU codes.
As the code words are not selected to satisfy any specific property, the set of code words might not satisfy the rectangularity property. Hence the code obtained need not be a rectangular. □

4.8 Examples

The codes listed in Appendix A, are examples of codes using the structured distance approach. For the purpose of illustrating the scheme for structuring of the distances to obtain this class of codes, in this section, some examples have been worked out in details.

Example 4.8.1 *Consider a code search for (2-PSK, 4-PSK, 4, 16, $\sqrt{8}$) BCM scheme.*

The channel signal constellation is defined by the set of Euclidean distances for the 4-PSK signal constellation,
$S = \{ d_0 = 0,\ d_1 = \sqrt{2},\ d_2 = 2 \}$.
The set of valid distances for the distance distribution of the code is,
$\hat{D}_{S' \times S' \times S' \times S'} = \{ d_1 d_1 d_1 d_1,\ \tilde{S}_{d_2 d_1 d_1 d_0},\ \tilde{S}_{d_2 d_1 d_1 d_1},\ \tilde{S}_{d_2 d_2 d_0 d_0},\ \tilde{S}_{d_2 d_2 d_0 d_1},\ \tilde{S}_{d_2 d_2 d_1 d_1},\ \tilde{S}_{d_2 d_2 d_2 d_1},\ \tilde{S}_{d_2 d_2 d_2 d_0},$
$d_2 d_2 d_2 d_2 \}$.

Where, for example, the set $\tilde{S}_{d_2 d_1 d_1 d_0}$, is the set of all the permutations of the distance 4-tuple,

$\tilde{S}_{d_2 d_1 d_1 d_0} = \{\, d_2d_1d_1d_0,\ d_2d_1d_0d_1,\ d_2d_0d_1d_1,\ d_0d_2d_1d_1,\ d_0d_1d_2d_1,\ d_0d_1d_1d_2,\ d_1d_0d_1d_2,\ d_1d_1d_0d_2,\ d_1d_1d_2d_0,\ d_1d_2d_0d_1,\ d_1d_2d_1d_0,\ d_1d_0d_2d_1 \,\}$.

These permutations, give code words at the same Euclidean distance

$$d_{\min} = \sqrt{d_2^2 + d_1^2 + d_1^2 + d_0^2} = \sqrt{8},$$

from some other code word.

Only those permutations are selected which when composed give valid distances.

Since, $d_2d_1d_1d_0$ and $d_2d_1d_0d_1$ are valid, and,

$d_2 \circ d_2\ d_1 \circ d_1\ d_1 \circ d_0\ d_0 \circ d_1 = d_0d_2d_1d_1$ is also a valid distance, hence, the distance 4-tuples $d_2d_1d_1d_0$ and $d_2d_1d_0d_1$ can be used in the distribution.

But, $d_2d_2d_2d_1$ and $d_2d_2d_1d_2$ are valid, but,

$d_2 \circ d_2\ d_2 \circ d_2\ d_2 \circ d_1\ d_1 \circ d_2 = d_0d_0d_1d_1$ is not a valid distance, and hence, the distance 4-tuples $d_2d_2d_2d_1$ and $d_2d_2d_1d_2$ can not be used in the distribution.

If a distance 4-tuple say $d_1\, d_1\, d_1\, d_1$, is considered with an initial code word 0000, then the signal set $\{\, 1, 3 \,\}$ is at the required distance.

If all the valid permutations of $\{\, 1, 3 \,\}$, that is, 4-tuples of the form 1111, 3333, 1133, 3311, 1331. 3131, 3331, 3313, ... etc. are considered. And, if 0000, 1111, 3333 are valid code words, then 3331 can not be selected as a code word since the distance between 3333 and 3331, corresponds to,

$d_1 \circ d_1\ d_1 \circ d_1\ d_1 \circ d_1\ d_1 \circ d_1 = d_0d_0d_0d_2 \notin \hat{D}_{S' \times S' \times S' \times S'}$.

The final set of valid code words obtained by this approach is,

$C = \{$ 0000, 1111, 3333, 2222, 3311, 1133, 1331, 3113, 1313, 3131, 2200, 0022, 2002, 0220, 0202, 2020 $\}$.

The encoder for this code, can be obtained from Table 4.8.1, as explained in Section 3.8. The mapping between the signals of the expanded channel signal constellation and the binary representation is as shown in Table 4.8.2.

Example 4.8.2 *Consider a code search for (2-PSK, 5-PSK, 4, 18, $\sqrt{5}$) BCM scheme.*

Table 4.8.1: Mapping from symbols to binary for code in Example 4.8.1

Data word	Binary data word ($x_3x_2x_1x_0$)	Code word ($z_3z_2z_1z_0$)	Binary code word ($y_7y_6y_5y_4y_3y_2y_1y_0$)
0	0000	0000	00000000
1	0001	1111	01010101
2	0010	3333	11111111
3	0011	2222	10101010
4	0100	3311	11110101
5	0101	1133	01011111
6	0110	1331	01111101
7	0111	3113	11010111
8	1000	1313	01110111
9	1001	3131	11011101
10	1010	2200	10100000
11	1011	0022	00001010
12	1100	2002	10000010
13	1101	0220	00101000
14	1110	0202	00100010
15	1111	2020	10001000

Table 4.8.2: Mapping from binary representation to channel symbols for the code in Example 4.8.1

Binary representation	4-PSK Channel signal
00	0
01	1
10	2
11	3

The channel signal constellation is defined by the set of Euclidean distances for the 5-PSK signal constellation,
$S = \{\, d_0 = 0,\ d_1 = \sqrt{1.38},\ d_2 = \sqrt{3.62}\,\}$.
The set of valid distances for the distance distribution of the code is,
$\hat{D}_{S'\times S'\times S'\times S'} = \{\, d_1d_1d_1d_1,\ \tilde{S}_{d_2d_1d_1d_0},\ \tilde{S}_{d_2d_1d_1d_1},\ \tilde{S}_{d_2d_2d_0d_0},\ \tilde{S}_{d_2d_2d_0d_1},\ \tilde{S}_{d_2d_2d_1d_1},\ \tilde{S}_{d_2d_2d_2d_1},\ \tilde{S}_{d_2d_2d_2d_0},$
$d_2d_2d_2d_2\,\}$.

The final set of valid code words obtained by this approach is,
$C = \{$ 0022, 0220, 2200, 2002, 2020, 0202, 2222, 1133, 1331, 3311, 3113, 1313, 3131, 1111, 0044, 4400, 3344, 4433 $\}$.
The encoder for this code, can be obtained from Table 4.8.3, as explained in Section 3.8. The

Table 4.8.3: Mapping from symbols to binary for code in Example 4.8.2

Data word	Binary data word $(x_4x_3x_2x_1x_0)$	Code word $(z_3z_2z_1z_0)$	Binary code word $(y_{11}y_{10}y_9y_8y_7y_6y_5y_4y_3y_2y_1y_0)$
0	00000	0022	000000010010
1	00001	0220	000010010000
2	00010	2200	010010000000
3	00011	2002	010000000010
4	00100	2020	010000010000
5	00101	0202	000010000010
6	00110	2222	010010010010
7	00111	1133	001001011011
8	01000	1331	001011011001
9	01001	3311	011011001001
10	01010	3113	011001001011
11	01011	1313	001011001011
12	01100	3131	011001011001
13	01101	1111	001001001001
14	01110	0044	000000100100
15	01111	4400	100100000000
16	10000	3344	011011100100
17	10001	4433	100100011011

mapping between the signals of the expanded channel signal constellation and the binary representation is as shown in Table 4.8.4.

4.9 Concluding Remarks

A new class of codes, known as **codes based on the selective permutations of distances**, is proposed. The scheme for obtaining this class of codes uses the structured distance approach. The developed scheme can work with an arbitrary expanded channel signal

Table 4.8.4: Mapping from binary representation to channel symbols for the code in Example 4.8.2

Binary representation	5-PSK Channel signal
000	0
001	1
010	2
011	3
100	4

constellation.

The analogy between the problem of sphere packings and the problem of obtaining codes for BCM schemes is discussed. Considerations and properties of the codes, which follow from some results in sphere packings, are presented.

A structure, based on the results from sphere packings, is enforced on the Euclidean distances, to obtain the distance distribution of the code. This scheme is known as the selective permutations of distances. This basically gives a rule for the selection of distances in the process of obtaining the distance distribution of a code. The algorithm summarizing this scheme and some properties of this class of codes are given. Some examples have been worked out in detail, to illustrate the proposed scheme.

Chapter 5

Reduced Tree Based Soft Decoding for BCM Schemes

5.1 Introduction

This chapter proposes a scheme for the soft decoding of general block codes, used with BCM. The block codes obtained by the structured distance approach in Chapters 3 and 4 need not have linear, cyclic, group, GU, rectangular, etc. structure on the code words. For such general codes, soft decoding can be performed using the presented scheme.

In this chapter[1], the code tree, instead of the trellis, is used for the representation of the block code words. A reduced tree, to be used by the soft decoder, is obtained from the code tree. Parallel implementation can be achieved for a reduced tree based soft decoder. It also eliminates the back tracking required in a trellis based soft decoder using the Viterbi algorithm. The proposed scheme is particularly well suited for the fast soft decoding, of general block codes, of short block lengths.

The chapter begins with a brief overview of some essential terms and notations. The code tree and its use as a representation of block codes is illustrated. Various techniques for reducing the code tree are presented. Considerations for the parallel implementation of the soft decoders are discussed. An algorithm summarizes the procedure for soft decoding. Reduced tree based soft decoding, for the example codes of Chapters 3 and 4, is illustrated.

[1]The scheme proposed in this chapter was partly presented at the **"National Conference on Communications–1996"** [44].

Finally, the chapter ends with some concluding remarks.

This chapter examines the use of a reduced tree for soft decoding. Obtaining the minimal trellis, for a Viterbi algorithm based soft decoder, is discussed in Chapter 6.

5.2 Background and Preliminaries

In a communication system, at the receiving end of the channel, the demodulator is followed by a decoder[2]. The decoder is used to obtain the data words from the received words, after error correction. For coded modulation and hence for BCM it is necessary to use a soft decoder. The soft decoder uses all the received information and makes the decision regarding the received word.

For AWGN channels the maximum likelihood receiver is identical to the minimum Euclidean distance receiver [16]. Hence, the soft decoder is basically a minimum Euclidean distance decoder.

The mapping between the data words and the code words for a BCM scheme is an, one-to-one and on-to map. For the purpose of decoding (obtaining the data words from the code words) an inverse map always exists. Primarily, the soft decoder computes the Euclidean distance of the received word[3] from all the code words, and then selects the code word at the minimum Euclidean distance as the received word, which is mapped to the corresponding data word. Basically, a soft decoder requires the following.

(1) Representation of all the code words.

(2) Computation of the Euclidean distances of the received word with all the code words.

(3) Storage of the Euclidean distances between the received word and all the code words.

(4) Comparison of the Euclidean distances to obtain the minimum Euclidean distance.

(5) Map from the decoded word to the data word.

[2]Refer Figure 1.1.3.

[3]The received word is a sequence of channel symbols with AWGN.

Hence, for implementation of a soft decoder the following requirements have to be satisfied.

(1) A scheme for the representation of the code words, which can be efficiently utilized by the other blocks of the decoder.

(2) A processing unit for computations of Euclidean distances between signals.

(3) Memory for storing the Euclidean distances, between the received word and the code words, which are real numbers.

(4) A processing unit for performing comparisons of real numbers.

(5) The code table for obtaining the data words from the decoded word.

As explained in Section 1.1, these blocks for a soft decoder are implemented in a modem.

TCM uses convolutional codes. For the soft decoding of TCM, a trellis is used for the representation of the code and the decoder is implemented using the Viterbi algorithm. As coded modulation originated from TCM and then schemes using BCM were later introduced, generally, for BCM too soft decoding is performed, as with TCM. Hence in most of the references on BCM schemes a trellis is obtained for the block code and the Viterbi algorithm is used for soft decoding. This chapter proposes a different scheme for soft decoding which is more suitable for BCM.

5.3 Motivating Factors

(1) Convolutional code consists of a stream of symbols. These are sequentially received one symbol at each stage. So the decoding also sequentially proceeds step by step. Hence, the trellis representation and the Viterbi algorithm, which does stage by stage decoding is suitable. On the other hand, block codes are of a finite shorter length. A scheme is required which can use the full received block and decode the block, instead of sequential symbol by symbol decoding. Such a decoder is more amenable to parallel implementation. For this, the trellis is not an appropriate representation scheme.

(2) For channels with large bandwidth[4], the rate of symbol transmission is high. Hence the modem has to be a very fast device. In such modems it is observed that, from the point of view of coded modulation, the primary bottleneck is the soft decoder. Hence, schemes are required to perform fast soft decoding. This is the important motivating factor for the search for general (non-linear) codes of short lengths over arbitrary channel signal constellations, considered in Chapters 3 and 4. It is this factor which has also motivated the scheme developed later on in Chapter 7.

(3) For the general codes obtained in Chapters 3 and 4, which might not be linear, cyclic, group, GU, rectangular, etc. some scheme is required for representation of the code words for soft decoding.

These factors emphasize the necessity for a different representation of the code words for a BCM scheme. Basically, a trellis is obtained from a tree [86]. Compared to trellis, the tree provides a better scope for utilization of parallelism. Hence, it is considered to use a code tree for soft decoding of codes for BCM. Various issues involved in using the code tree for soft decoding of block codes used in BCM schemes are discussed in the following sections of this chapter.

5.4 The Code Tree

The code tree [86] is a structure used for a representation of the code. This chapter develops the necessary frame work for an efficient representation of all the code words of a BCM scheme using the code tree.

The code tree to be used for the representation of code words for a BCM scheme can be characterized by the following properties.

(1) The code tree for a BCM scheme, is a weighted n'-ary tree, where n' is the number of channel signals in the expanded channel signal constellation.

(2) The weights, or labels associated with the edges of the tree are the channel symbols from S'.

[4] For example television cables.

(3) The depth of the code tree is equal to n, the block length of the code.

(4) The total number of vertices at the final level n is equal to the number of the code words $|C|$. Hence, each vertex at level n is associated with an unique code word.

(5) A code word is represented by a path in the tree starting from the root vertex and ending at a vertex at the final level n.

(6) The code tree representation for a block code is unique up to isomorphisms.

For soft decoding, it is necessary to compute the Euclidean distances of a received word with the code words and store these distances for comparison. In the tree representation of a code, these Euclidean distances are computed using the edges of the tree along various paths and are known as path metrics. The path metrics along various paths are stored at the vertices of the tree.

Definition 5.4.1 *For a code tree, at a level* i, $0 \leq i \leq n$, $v_{i,j}$ *represents a vertex of the tree at level* i.
$v_{0,0}$, *is the root vertex at level* 0.
At level n, $v_{n,j}$ $0 \leq j \leq |C|$, *are the vertices uniquely representing the code words.*

Definition 5.4.2 *The code tree* $\mathbf{T} = \mathbf{T}_{v_{0,0}}$, *is the set of all the paths from the root vertex to all the vertices at level* n.

Hence, the code tree is a representation for all the code words C, of a BCM scheme.

Definition 5.4.3 *The subtree from a vertex* j *at level* i, *denoted as* $\mathbf{T}_{v_{i,j}}$, *is the set of paths starting from this vertex* $v_{i,j}$ *and terminating at the vertices at level* n.

Definition 5.4.4 *The path metric, denoted by* $d_{v_{i,j}}$, *along a path through the vertex* $v_{i,j}$ *is stored at the vertex* $v_{i,j}$, $0 \leq i \leq n$. *The distances stored at vertices from level* 1 *to level* $n-1$, *are known as partial path metrics. The distances at the vertices at level* n *are the final path metrics or the Euclidean distances between the received word and all the code words.*

Example 5.4.1 Consider the block code for the (2-PSK, 4-PSK, 3, 8, $\sqrt{6}$) BCM scheme obtained in Example 3.6.4.

The code words are $C = \{000, 012, 132, 120, 321, 201, 213, 333\}$. The code tree for this code is given in Figure 5.4.1. There are $|C| = 8$ vertices at level $n = 3$ of the code tree. The

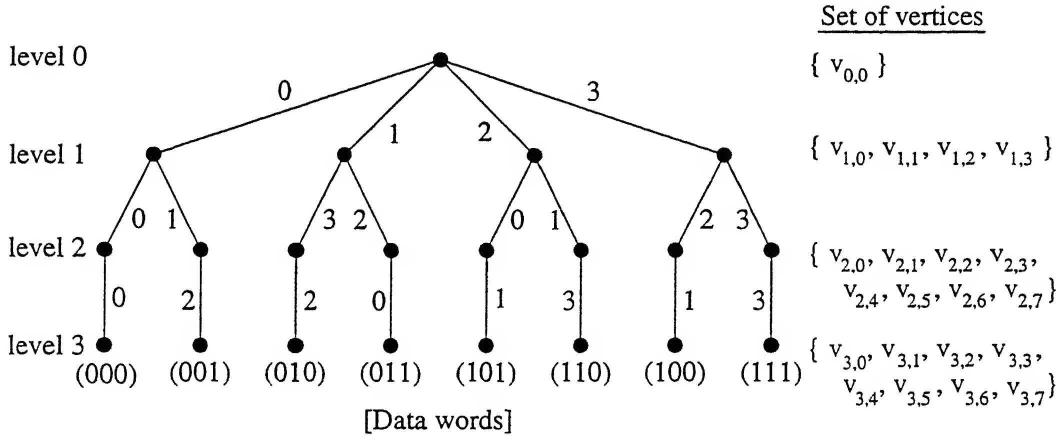

Figure 5.4.1: Code tree representation for the code in Example 5.4.1

subtree $T_{v_{2,0}}$, starting from vertex $v_{2,0}$ is illustrated in Figure 5.4.2.

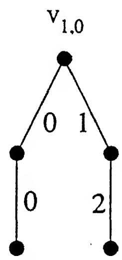

Figure 5.4.2: Example of a subtree of the code tree for the code in Example 5.4.1

5.4.1 Soft Decoding Using the Code Tree

The code tree representation for the code words can be used in the implementation of a soft decoder for a BCM scheme. This is discussed in brief over here. A more detailed discussion is given for the soft decoding using the reduced tree.

The Euclidean distances between the received channel symbols and the edges of the code tree are computed. The partial path metrics are stored at the vertices from level 1, to level $n-1$. The final path metrics at the vertices at level n corresponding to the Euclidean distance of the received word with all the code words, are found. Once the metrics at all the vertices at level n have been computed, a comparison of the metrics is necessary to obtain the minimum. The path in the code tree corresponding to that final vertex, with the minimum metric, is the decoded word. In fact, with each final vertex a data word can be assigned, as there is a one to one correspondence between the data word and the vertices at level n. Hence, back tracking is not required. Also, since a processing element has to compute Euclidean distances, add, store and compare. In the various branches of the tree the processing can take place in parallel.

The main disadvantage of the tree representation is that the number of vertices, and hence the storage required for data, grows rapidly with the size of the code. To reduce this growth, and make the tree suitable for representation of large block codes, it is necessary to reduce the code tree obtained for a BCM scheme. This is discussed in the next section of this chapter.

5.5 Code Tree Reduction

Reduction of the code tree for the soft decoding of codes for BCM scheme has to be carried out to reduce the storage requirements. This reduction of the storage is achieved at the cost of reduction in the parallelism of the tree. In this way a trade off exists between parallelism and memory storage for a tree based soft decoder. Depending on the specific need of the application, a proper configuration of the soft decoder has to be designed.

To reduce the storage requirements for the code tree, it is necessary to eliminate as many vertices from consideration at lower levels in the code tree as possible.

Definition 5.5.1 *The elimination of a vertex $v_{i,j}$ of a tree from further consideration, implies that, the subtree rooted at that vertex $\mathbf{T}_{v_{i,j}}$ is eliminated.*

When a vertex is eliminated from further consideration, no computations are performed for the subtree rooted at that vertex.

The decision about reducing the tree is made during the design of the soft decoder, but the actual reduction due to the elimination of vertices from further consideration takes place during the decoding of a received word. The elimination of the vertices at the lower levels result in a reduction of the vertices at the final level. The storage required for the decoder is reduced due to an overall reduction in the number of vertices of the tree.

5.5.1 Reduction of Subtrees Isomorphic with Weight

Some of the subtrees in a tree might be redundant from the point of view of soft decoding. Detection of these redundant subtrees and their deletion can lead to the reduction of the tree [5] To remove a subtree it is necessary to eliminate the vertex at which the subtree is rooted.

The lower the level at which elimination of a vertex from further consideration is possible the better it is, since this leads to an elimination of a large number of vertices of the tree. Hence, starting from level 1 to level $(n-1)$, subtrees rooted at the vertices are considered.

Definition 5.5.2 *For $i = 0$ to $i = n-1$, consider the set of all the vertices at level i, denoted by $\{ v_{i,j} \}$.*
The cardinality of $\{ v_{i,j} \} = l_i$.

Theorem 5.5.1 *For a BCM scheme with block code C, $l_0 = 1$, $l_n = |C|$ and $l_i \leq |C| \; \forall \, i$, such that, $1 \leq i \leq n-1$.*

Proof: There is only the root vertex $v_{0,0}$ at level 0. Hence, $l_0 = 0$.
From Definitions 5.4.1 and 5.4.2, the number of vertices at level n is $|C|$. Hence, $l_n = |C|$.
$l_i \leq |C| \; \forall \; i$, such that $1 \leq i \leq n-1$, since $l_n = |C|$. □

Definition 5.5.3 *Let, $\mathrm{V}_{i,k}$ be a partition set, such that, $\forall \; k$, $0 \leq k \leq l_i - 1$ $\mathrm{V}_{i,k} \subset \{ v_{i,j} \}$. Then,*

$$\bigcup_{\forall k} \mathrm{V}_{i,k} = \{ v_{i,j} \} \textit{ and}$$

$$\mathrm{V}_{i,k_1} \bigcap_{k_1 \neq k_2,\; 0 \leq k_1, k_2 \leq l_i - 1} \mathrm{V}_{i,k_2} = \emptyset.$$

[5]The reduction of code tree has appeared in the *IEE Proc. Commun.*, April 1997 [43].

Definition 5.5.4 *The elements of the set* $\mathbf{V}_{i,k}$ *are such that,* $v_{i,j_1} \in \mathbf{V}_{i,k}$ *and* $v_{i,j_2} \in \mathbf{V}_{i,k}$ *for some* $0 \leq j_1, j_2 \leq l_i - 1$,

$$iff \quad \mathbf{T}_{v_{i,j_1}} \overset{w}{\cong} \mathbf{T}_{v_{i,j_2}},$$

where, $\overset{w}{\cong}$ *stands for isomorphic with weights. The subtrees* $\mathbf{T}_{v_{i,j-1}}$ *and* $\mathbf{T}_{v_{i,j-2}}$*, will be known as subtrees isomorphic with weights.*

Theorem 5.5.2 *If the cardinality of* $\mathbf{V}_{i,k} > 1$*, for some* k*, then the elements of* $\mathbf{V}_{i,k}$*, that is the vertices* $v_{i,j}$ *of the code tree can be eliminated from further considerations for soft decoding and* $\mathbf{RV}_{i,k} \subset \mathbf{V}_{i,k}$ *can be obtained, such that,* $\forall\ k$*, cardinality of* $\mathbf{RV}_{i,k} = 1$*, where* $\mathbf{RV}_{i,k}$ *is the partition set at level* i *of the reduced tree.*

Proof: Reduction of a tree is only possible if,
$\exists\ k = k'$ such that $0 \leq k' \leq l_i - 1$ and cardinality of $\mathbf{V}_{i,k'} > 1$.

CASE 1:
If the cardinality of $\mathbf{V}_{i,k} = 1$, for some k.
Then, no elimination is possible.
$\mathbf{V}_{i,k} = \mathbf{RV}_{i,k}$ and cardinality of $\mathbf{RV}_{i,k} = 1$, for those k.

CASE 2:
If cardinality of $\mathbf{V}_{i,k} > 1$,
then, let cardinality of $\mathbf{V}_{i,k} = m_{i,k}$, where, $m_{i,k} > 1$.
From Definition 5.5.3,

$$\mathbf{V}_{i,k} \subset \{v_{i,j}\} = \{\, v_{i,j_0},\ v_{i,j_1},\ \ldots\ v_{i,j_{m_{i,k}-1}}\,\}.$$

From Definition 5.5.4, it follows that,

$$\mathbf{T}_{v_{i,j_0}} \overset{\mathrm{W}}{\cong} \mathbf{T}_{v_{i,j_1}} \overset{\mathrm{W}}{\cong} \ldots \overset{\mathrm{W}}{\cong} \mathbf{T}_{v_{i,j_{m_{i,k}-1}}}. \tag{5.5.1}$$

The path metrics for the paths from the root vertex to the vertices in the set $\mathbf{V}_{i,k}$, form a set,

$$\mathbf{D}_{i,k} = \{\, d_{v_{i,j_0}},\ d_{v_{i,j_1}},\ \ldots,\ d_{v_{i,j_{m_{i,k}-1}}}\,\}.$$

Consider the poset, $\mathbf{P}_{i,k} = \{\, \mathbf{D}_{i,k} \leq \,\}$.
The set of minimal elements of $\mathbf{P}_{i,k}$ is $\mathbf{MP}_{i,k} = \{\, d_{v_{i,j_q}} \,\}$, for some $\{\, q \,\}$, such that, $0 \leq q \leq m_{i,k} - 1$.

Statement 1:

If the cardinality of $\mathbf{MP}_{i,k} = 1$, then the least element of the poset is, $lP_{i,k} = d_{v_{i,j_{q'}}}$ for some $q = q'$.
If the cardinality of $\mathbf{MP}_{i,k} > 1$, then the least element of the poset is, $lP_{i,k} = d_{v_{i,j_{q'}}}$ for q' any arbitrary element of $\{q\}$.

From Equation 5.5.1, it follows that

$$\{v_{n,j0_p}\} \overset{\mathrm{W}}{\cong} \{v_{n,j1_p}\} \overset{\mathrm{W}}{\cong} \ldots \overset{\mathrm{W}}{\cong} \{v_{n,j{m_{i,k}-1}_p}\} \tag{5.5.2}$$

where, $1 \leq p \leq p'$ and p' = the number of vertices of the subtree at level n of the tree. From Equations 5.5.1 and 5.5.2 it follows that,

$$\{d(v_{n,j0_p} - v_{i,j0})\} = \{d(v_{n,j1_p} - v_{i,j1})\} = \cdots = \{d(v_{n,j(m_{i,k}-1)_p} - v_{i,j(m_{i,k}-1)})\} \;\forall p \tag{5.5.3}$$

where $d(v_y - v_x)$ denotes the sum of path metrics for the partial path from the vertex at level x to the vertex at level y, assuming that $d_{v_x} = 0$.
From Equation 5.5.1, Statement 1, Equations 5.5.2 and 5.5.3, it follows that,

$$\mathbf{MP}_{n,k} \subset \{d_{v_{n,j_{q'_p}}}\} \quad \forall p.$$

Using argument similar to Statement 1,

$$lP_{n,k} \in \{d_{v_{n,j_{q'_p}}}\}, \text{ for some } p.$$

All the other elements of the poset $\mathbf{P}_{n,k}$ will be greater. It is of interest to obtain the least $d_{v_{n,j}}$.
So all the other elements of $\mathbf{V}_{i,k}$, but for $v_{i,j_{q'}}$ can be eliminated from further consideration in soft decoding to obtain $\mathbf{RV}_{i,k}$, such that,

$$\mathbf{RV}_{i,k} = \{\, v_{i,j_{q'}} \,\} \text{ is a singleton.}$$

So the cardinality of $\mathbf{RV}_{i,k} = 1$.

□

From the theorem it can be inferred that, if some

$$\mathbf{V}_{i,k} = \{ v_{i,j_0}, v_{i,j_1}, \ldots v_{i,j_{m_{i,k}-1}} \}.$$

Then it is necessary to compute,

$$\mathbf{D}_{i,k} = \{ d_{v_{i,j_0}}, d_{v_{i,j_1}}, \ldots, d_{v_{i,j_{m_{i,k}-1}}} \}.$$

The elements of this set $\mathbf{D}_{i,k}$ have to be compared and the minimum element has to be found.
Say, it is $d_{v_{i,j_{q'}}}$.
Then only the vertex $d_{v_{i,j_{q'}}} \in \mathbf{D}_{i,k}$ is kept and all the other vertices are eliminated from further consideration for soft decoding.
The reduced tree will only have $\mathbf{RV}_{i,k} = \{ v_{i,j_{q'}} \}$ and the path through this vertex.
All the other vertices of $\mathbf{V}_{i,k}$, and the paths of the tree through them will be eliminated for the purpose of soft decoding.
Note that the decision that a reduction is to be performed, $m_{i,k} > 1$ is done during the design of soft decoder.
But, the comparison of distances to find the minimum element of $\mathbf{D}_{i,k}$, is performed during the actual decoding of a received word.

Based on this theorem, subtrees in the tree can be identified, which can be deleted to achieve reduction of the tree. As no assumptions regarding the nature of the code are made the scheme is valid for general codes. This theorem is also the basis for obtaining minimal proper trellises for general codes, discussed in details in Chapter 6. The storage space required for the data and the number of comparisons required are same for a trellis and for the reduced tree.

Example 5.5.1 Consider the code tree obtained in Example 5.4.1.

At level 0, only the root vertex is present and no reduction is possible.
$\mathbf{V}_{0,0} = \mathbf{RV}_{0,0} = \{ v_{0,0} \}$.
At level 1, the code tree has 4 vertices. The subtrees $\mathbf{T}_{v_{1,0}}$, $\mathbf{T}_{v_{1,1}}$, $\mathbf{T}_{v_{1,2}}$ and $\mathbf{T}_{v_{1,3}}$ are shown in Figure 5.5.1. Since these subtrees are not isomorphic with weight, no reduction is possible at this level.

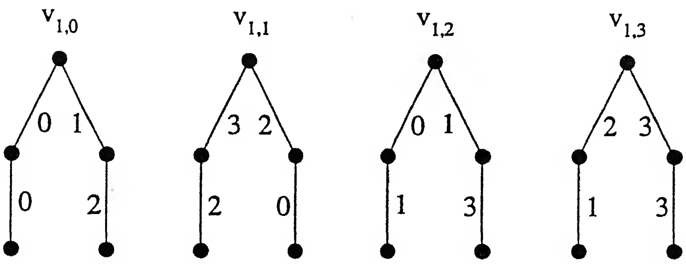

Figure 5.5.1: The subtrees $T_{v_{1,0}}$, $T_{v_{1,1}}$, $T_{v_{1,2}}$ and $T_{v_{1,3}}$ for the code in Example 5.5.1

$V_{1,0} = RV_{1,0} = \{ v_{1,0} \}$,
$V_{1,1} = RV_{1,1} = \{ v_{1,1} \}$,
$V_{1,2} = RV_{1,2} = \{ v_{1,2} \}$ and
$V_{1,3} = RV_{1,3} = \{ v_{1,3} \}$.
At level 2, the code tree has 8 vertices. The subtrees $T_{v_{2,0}}$, $T_{v_{2,1}}$, $T_{v_{2,2}}$, $T_{v_{2,3}}$, $T_{v_{2,4}}$, $T_{v_{2,5}}$, $T_{v_{2,6}}$ and $T_{v_{2,7}}$ are shown in Figure 5.5.2.
$T_{v_{2,0}} \overset{W}{\cong} T_{v_{2,3}}$,

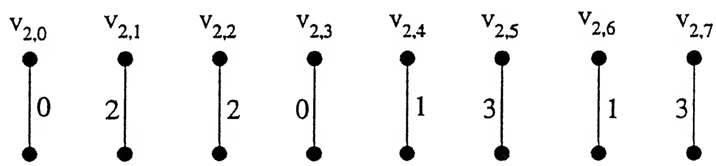

Figure 5.5.2: The subtrees $T_{v_{2,0}}$, $T_{v_{2,1}}$, $T_{v_{2,2}}$, $T_{v_{2,3}}$, $T_{v_{2,4}}$, $T_{v_{2,5}}$, $T_{v_{2,6}}$ and $T_{v_{2,7}}$ for the code in Example 5.5.1

$T_{v_{2,1}} \overset{W}{\cong} T_{v_{2,2}}$,
$T_{v_{2,4}} \overset{W}{\cong} T_{v_{2,6}}$ and
$T_{v_{2,5}} \overset{W}{\cong} T_{v_{2,7}}$.
Hence, reduction of the vertices is possible at this level.
$V_{2,0} = \{ v_{2,0}, v_{2,3} \}$,
$V_{2,1} = \{ v_{2,1}, v_{2,2} \}$,
$V_{2,2} = \{ v_{2,4}, v_{2,6} \}$ and

$\mathbf{V}_{2,3} = \{ v_{2,5}, v_{2,7} \}$.

Comparison of the elements of the following sets will have to be performed, during the decoding of a received word.

$\mathbf{D}_{2,0} = \{ d_{v_{2,0}}, d_{v_{2,3}} \}$,

$\mathbf{D}_{2,1} = \{ d_{v_{2,1}}, d_{v_{2,2}} \}$,

$\mathbf{D}_{2,2} = \{ d_{v_{2,4}}, d_{v_{2,6}} \}$ and

$\mathbf{D}_{2,3} = \{ d_{v_{2,5}}, d_{v_{2,7}} \}$.

Assume, that for a received word the partial metrics are as follows.

$d_{v_{2,0}} \leq d_{v_{2,3}}$,

$d_{v_{2,1}} \leq d_{v_{2,2}}$,

$d_{v_{2,4}} \leq d_{v_{2,6}}$ and

$d_{v_{2,5}} \leq d_{v_{2,7}}$.

Then, $\mathbf{RV}_{2,0} = \{ v_{2,0} \}$,

$\mathbf{RV}_{2,1} = \{ v_{2,1} \}$,

$\mathbf{RV}_{2,2} = \{ v_{2,4} \}$ and

$\mathbf{RV}_{2,3} = \{ v_{2,5} \}$.

At level 3 only four vertices $v_{3,0}$, $v_{3,1}$, $v_{3,4}$ and $v_{3,5}$ remain.

The reduced tree is shown in Figure 5.5.3.

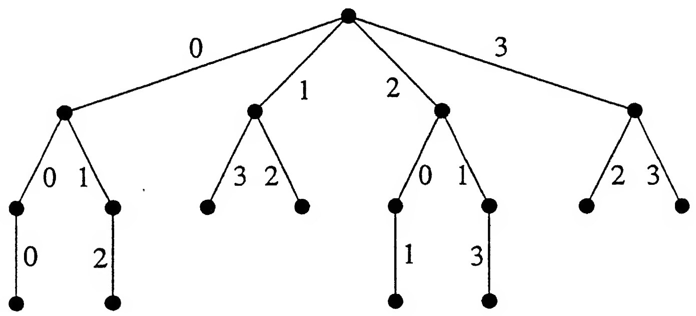

Figure 5.5.3: Reduced tree for the code in Example 5.5.1

5.5.2 Reduction Based on Code Equivalence

The code tree can be further reduced, by using an equivalent code, and then further employing the reduction technique proposed in the previous subsection. Equivalent codes have been

defined in Definition 3.3.6. Work has been done by Forney [26], Muder [61], Kasami [47,48] on finding efficient trellises for binary codes. In this section equivalent codes are used for further reducing the code tree of general block codes.

Theorem 5.5.3 *Equivalent codes have different code trees.*

Proof: The code tree is a weighted tree. The paths in the tree starting from the root vertex to the vertices at level n, represent the code words.
For an equivalent code the code word n-tuples are permuted, and are different. Hence, for the tree too the vertices and the weights, that is, the labels associated with the edges change. Hence, equivalent codes have different code trees. □

Lemma 5.5.1 *The reduced tree and trellis for equivalent codes are different.*

Proof: This is a direct consequence of Theorem 5.5.3. □

Based on this lemma, an equivalent code resulting in the maximum reduction of the vertices of the tree can be used in the soft decoder.

For an equivalent code, the n-tuple elements of the distance distribution are also permuted in their co-ordinate position, but the over all Euclidean distances between the code words remain the same. Hence, for decoding an equivalent code can be used. If an equivalent code is used by a soft decoder, two schemes are possible as explained below.

(1) The block encoder is not modified. Only the representation of the code words in the decoder and the received word at the receiver, are used in a different manner, to optimize the soft decoder. The permutation applied to the code word has also to be applied to the received word. The code table is not effected by permutations of the co-ordinate positions of the code word. But soft decoding can proceed only after the entire block code of length n is received.

(2) The block encoder is modified and designed for the equivalent code which optimizes the soft decoder. The encoder now uses the permuted code words in the code table. In this case, it is not necessary to wait for the entire block code words of length n, and decoding can proceed as soon as a symbol is received.

Example 5.5.2 Consider the following code for a BCM scheme.
$C_0 = \{000, 011, 020, 021, 102, 110, 112, 121, 122\ 200\ 201, 202\}$.

The code tree for this code is shown in Figure 5.5.4. For this code tree, no reduction is

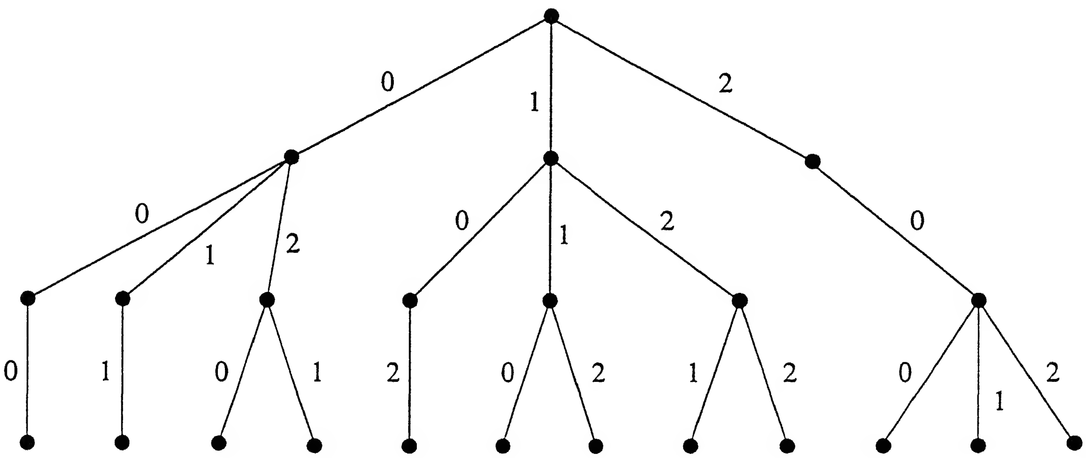

Figure 5.5.4: Code tree for C_0 in Example 5.5.2

possible at any levels. The number of vertices at each level are tabulated in Table 5.5.1. Consider the equivalent code C_1, obtained by permutation of the co-ordinate positions of

Table 5.5.1: Number of vertices for the reduced tree and code tree for C_0 in Example 5.5.2

Level	No. of vertices in code tree	No. of vertices in reduced tree
0	1	1
1	3	3
2	7	7
3	12	12

C_0.
$C_1 = \{000, 011, 002, 012, 120, 101, 121, 112, 122\ 200\ 210, 220\}$.
The code tree and the reduced code tree for this code are shown in Figures 5.5.5 and 5.5.6, respectively. For this code the number of vertices at each level of the code tree and the

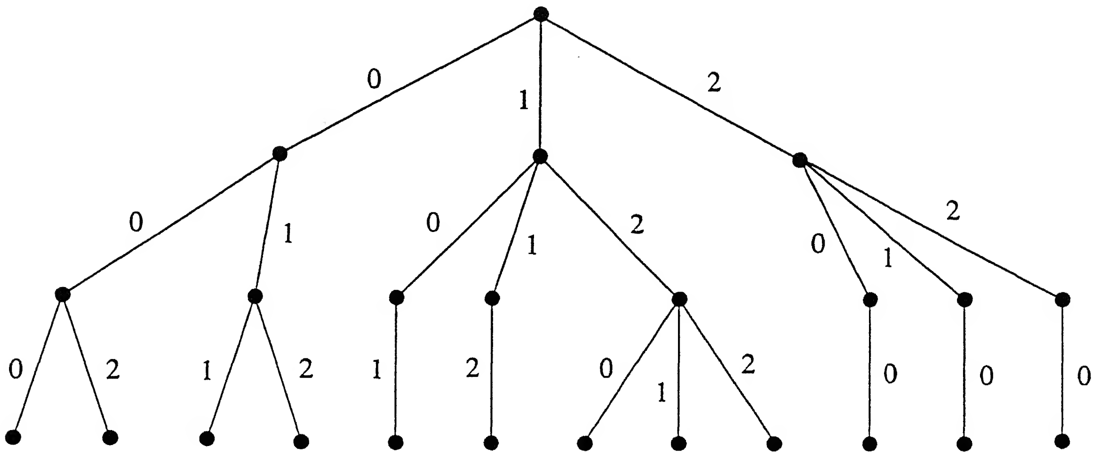

Figure 5.5.5: Code tree for C_1 in Example 5.5.2

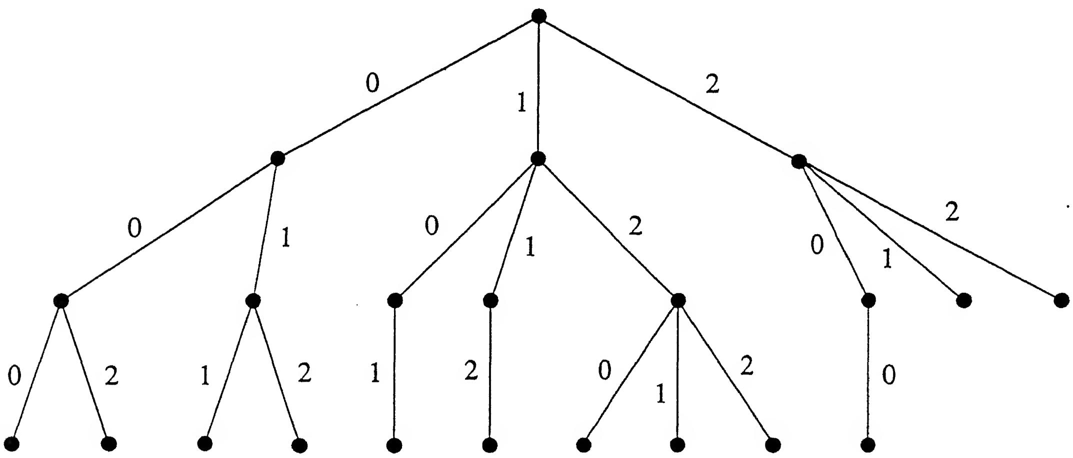

Figure 5.5.6: Reduced code tree for C_1 in Example 5.5.2

reduced tree are tabulated in Table 5.5.2. Consider the equivalent code C_2, obtained by permutation of the co-ordinate positions of C_0.

$C_2 = \{$ 000, 110, 200, 210, 021, 101, 121, 211, 221 002 012, 022 $\}$.

The code tree and the reduced code tree for this code are shown in Figures 5.5.7 and 5.5.8, respectively. For this code the number of vertices at each level of the code tree and the reduced tree are tabulated in Table 5.5.3. Consider the equivalent code C_3, obtained by permutation of the co-ordinate positions of C_0.

Table 5.5.2: Number of vertices for the reduced tree and code tree for C_1 in Example 5.5.2

Level	No. of vertices in code tree	No. of vertices in reduced tree
0	1	1
1	3	3
2	8	6
3	12	10

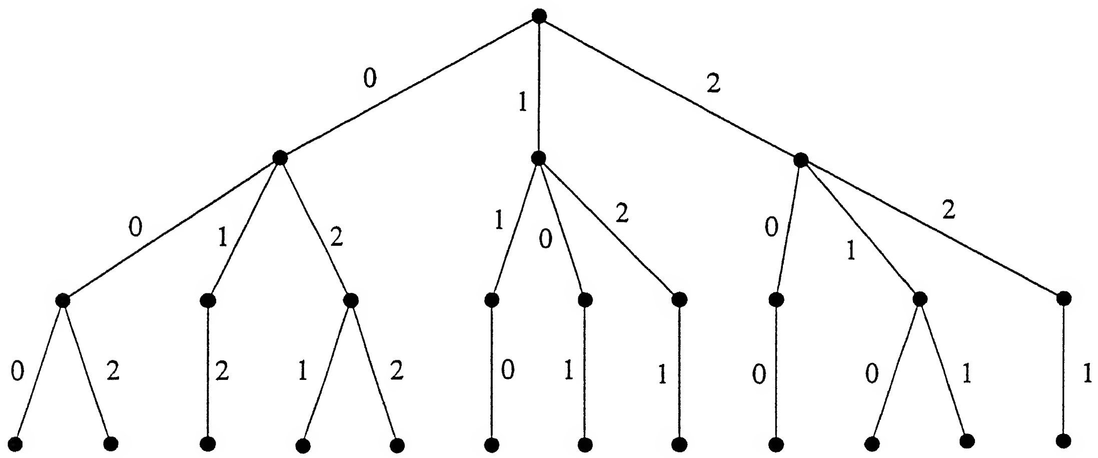

Figure 5.5.7: Code tree for C_2 in Example 5.5.2

Table 5.5.3: Number of vertices for the reduced tree and code tree for C_2 in Example 5.5.2

Level	No. of vertices in code tree	No. of vertices in reduced tree
0	1	1
1	3	3
2	9	6
3	12	9

$C_3 = \{\,000,\ 101,\ 200,\ 201,\ 012,\ 110,\ 112,\ 211,\ 212\ 020\ 021,\ 022\,\}$.

The code tree for this code is shown in Figure 5.5.9.

No tree reduction is possible for this code. For this code the number of vertices at each level

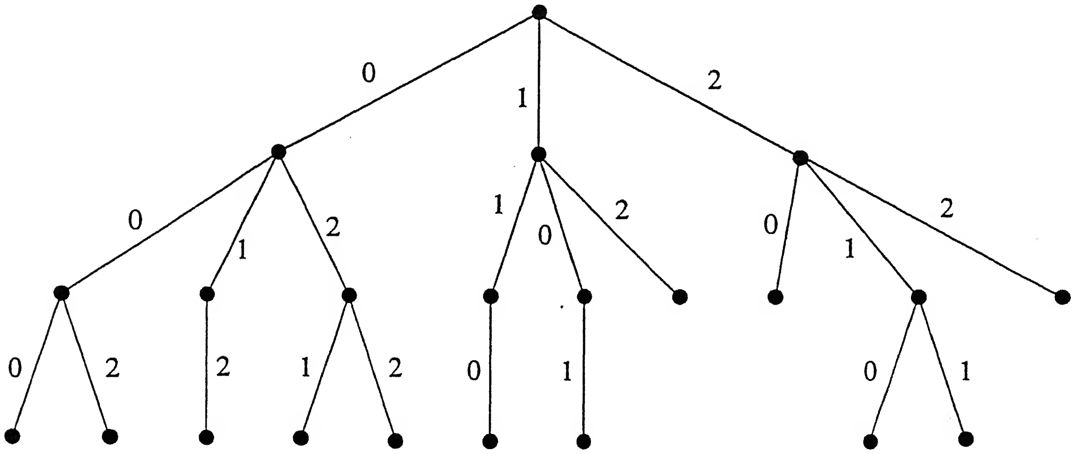

Figure 5.5.8: Reduced code tree for C_2 in Example 5.5.2

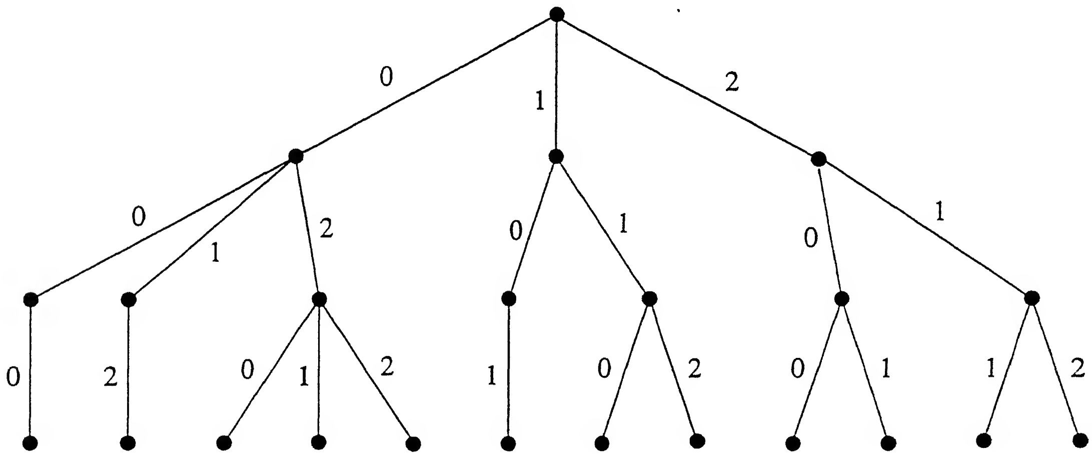

Figure 5.5.9: Code tree for C_3 in Example 5.5.2

of the code tree and the reduced tree are tabulated in Table 5.5.4. Consider the equivalent code C_4, obtained by permutation of the co-ordinate positions of C_0.
$C_4 = \{$ 000, 101, 002, 102, 210, 011, 211, 112, 212 020 120, 220 $\}$.
The code tree and the reduced code tree for this code are shown in Figures 5.5.10 and 5.5.11, respectively. For this code the number of vertices at each level of the code tree and the reduced tree are tabulated in Table 5.5.5. Consider the equivalent code C_5, obtained by permutation of the co-ordinate positions of C_0.

Table 5.5.4: Number of vertices for the reduced tree and code tree for C_3 in Example 5.5.2

Level	No. of vertices in code tree	No. of vertices in reduced tree
0	1	1
1	3	3
2	7	7
3	12	12

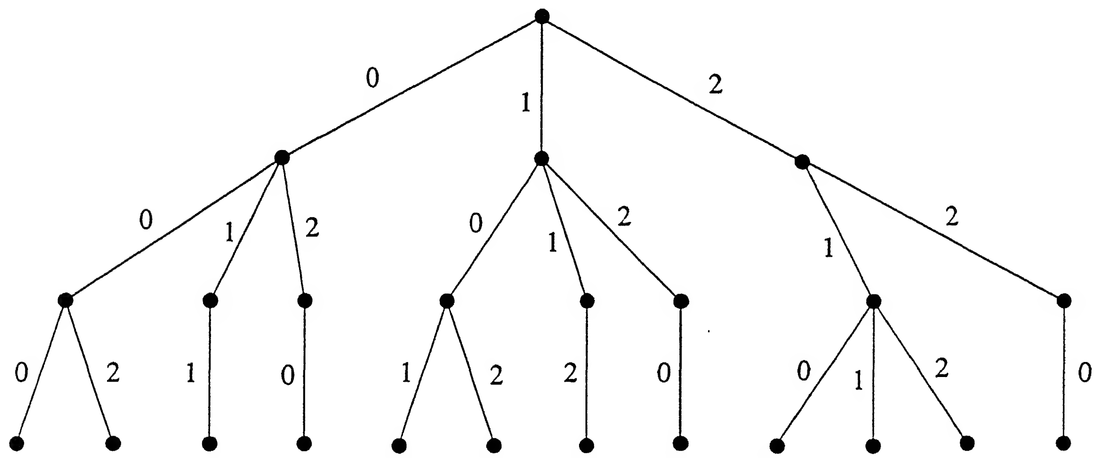

Figure 5.5.10: Code tree for C_4 in Example 5.5.2

Table 5.5.5: Number of vertices for the reduced tree and code tree for C_4 in Example 5.5.2

Level	No. of vertices in code tree	No. of vertices in reduced tree
0	1	1
1	3	3
2	8	6
3	12	10

$C_5 = \{\, 000,\ 110,\ 020,\ 120,\ 201,\ 011,\ 211,\ 121,\ 221\ 002\ 102,\ 202 \,\}$.
The code tree and the reduced code tree for this code are shown in Figures 5.5.12 and 5.5.13, respectively. For this code the number of vertices at each level of the code tree and the

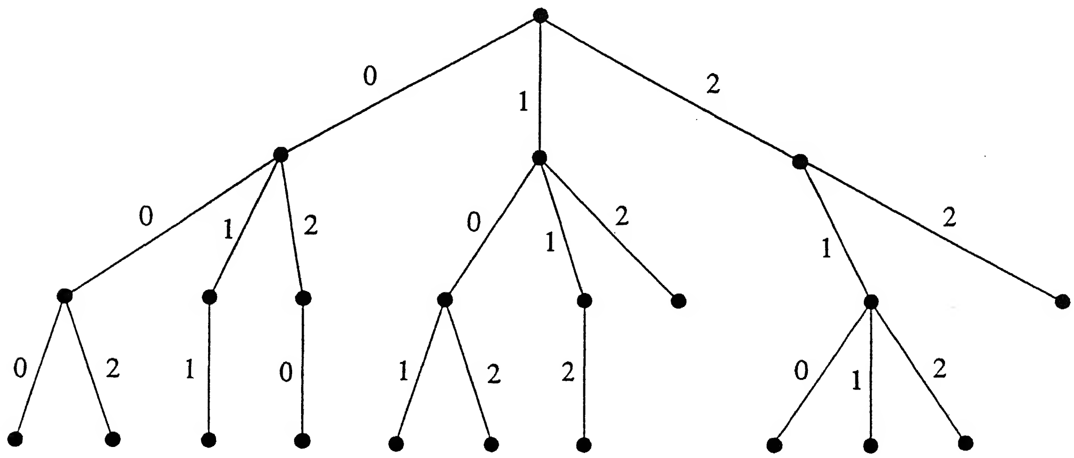

Figure 5.5.11: Reduced code tree for C_4 in Example 5.5.2

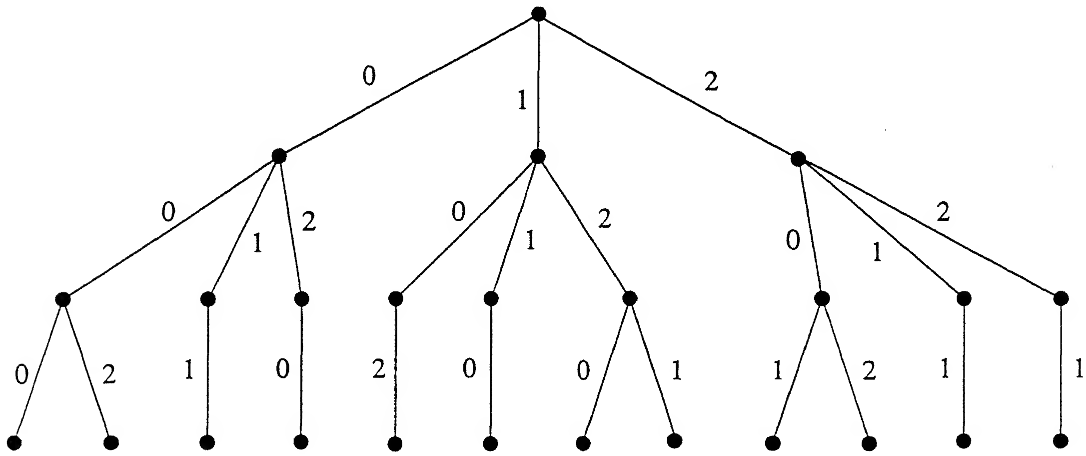

Figure 5.5.12: Code tree for C_5 in Example 5.5.2

reduced tree are tabulated in Table 5.5.6. From the tables, it is seen that the codes C_2 and C_5, results in the most reduction of vertices of the code tree. And these can be used for soft decoding. If $r_0r_1r_2$ is the received word, and if the code C_2 is used for a reduced tree based soft decoder, then the 3-tuple $r_1r_2r_0$ has to be considered, if the block encoder is not modified. If it is decided to modify the block encoder, then the encoder has to be designed for the code C_2 instead of C_0.

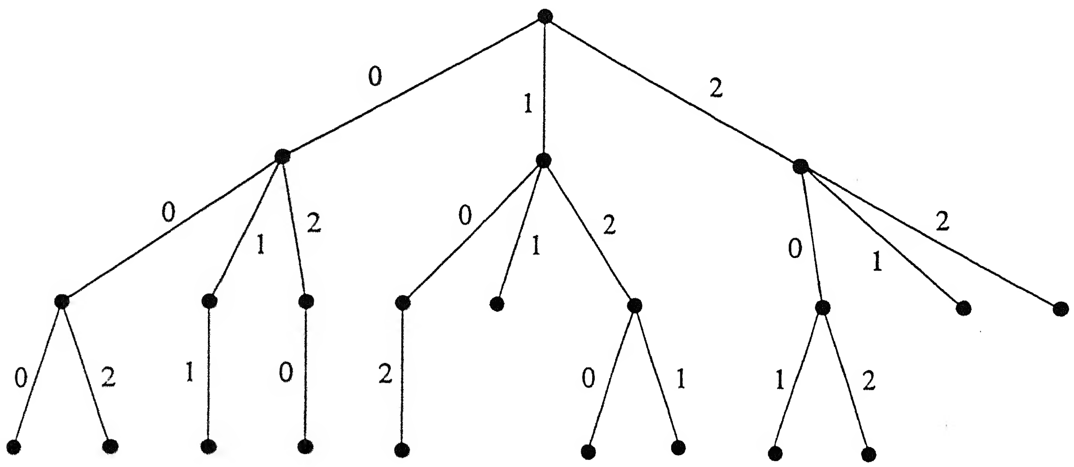

Figure 5.5.13: Reduced code tree for C_5 in Example 5.5.2

Table 5.5.6: Number of vertices for the reduced tree and code tree for C_5 in Example 5.5.2

Level	No. of vertices in code tree	No. of vertices in reduced tree
0	1	1
1	3	3
2	9	6
3	12	9

5.5.3 Reduction Based on Previously Computed Distance $d_{v_{n,j}}$

The partial path metric computed at any vertex $d_{v_{i,j}}$, has an upper bound. During soft decoding, if the metric exceeds the bound, then further processing in the subtree rooted at that vertex can be aborted. The possibility of this technique of further reduction, for a reduced tree, is recognized during the design of the soft decoder. Actual reduction might take place depending on the path metrics at the vertices during decoding.

Theorem 5.5.4 *During computation along any path, in the reduced tree, while soft decoding, if the partial path metric at any vertex $v_{i,j}$, $d_{v_{i,j}} \geq d_{v_{n,k'}}$ for some k' such that, $0 \leq k' \leq |C|-1$, and, $d_{v_{n,k'}}$ is the path metric of any vertex at the final level, of a path computed earlier to the path through $v_{i,j}$ in progress, then $v_{i,j}$ can be eliminated from further consideration for*

soft decoding.

Proof: Consider the set $\{ d_{v_{n,jp}} \}\ \forall\, p$, where p is the cardinality of the set of vertices at level n of the subtree rooted at $v_{i,j}$. $d_{v_{n,jp}}$ are the final path metrics of the vertices at level n, of the subtree $\mathbf{T}_{v_{i,j}}$, rooted at vertex $v_{i,j}$.

$$\forall\ x \text{ such that } x \in \{ d_{v_{n,jp}} \}, \text{ if, } d_{v_{i,j}} \geq d_{v_{n,k'}} \text{ then, } x \geq d_{v_{n,k'}}.$$

So, $\{ v_{n,j_p} \}$ will never be preferred over $v_{n,k'}$ for a soft decoded code word, since soft decoding selects a vertex with the minimum Euclidean distance.
This implies that, the vertex set $\{ v_{n,j_p} \}$ will be eliminated in the selection of a vertex corresponding to the decoded word.
Hence, $v_{i,j}$ can be eliminated from further considerations in soft decoding. □

The minimum metric of the vertex at the final level, upper bounds the computations of the partial metrics at all the remaining vertices of the tree. But this bound obtained is not an absolute upper bound for the computations at any vertex. An absolute upper bound to be used for tree reduction can be found for GU codes.

For GU codes, the Voronoi regions for all the codewords are congruent. The considerations which follow are applicable only for GU codes.

Definition 5.5.5 *Let, $\tilde{r}$ be the radius of the $\tilde{n}$-dimensional sphere centered at the code word circumscribing the Voronoi region.*

Theorem 5.5.5 *During computation along any path, in the reduced tree, during soft decoding for a GU code. If the partial path metric at any vertex $v_{i,j}$, $d_{v_{i,j}} > \tilde{r}$, then, $v_{i,j}$ can be eliminated from further consideration for soft decoding.*

Proof: Consider the set $\{ d_{v_{n,jp}} \}\ \forall\, p$, where p is the cardinality of the set of vertices at level n of the subtree rooted at $v_{i,j}$. $d_{v_{n,jp}}$ are the final path metrics of the vertices at level n, of the subtree $\mathbf{T}_{v_{i,j}}$, rooted at vertex $v_{i,j}$.

$$\forall\ x \text{ such that } x \in \{ d_{v_{n,jp}} \}, \text{ if, } d_{v_{i,j}} > \tilde{r} \text{ then, } x > \tilde{r}.$$

So, $\{ v_{n,j_p} \}$ will not be selected, as the received word will be in the Voronoi region of some other code word.

This implies that, the vertex set $\{ v_{n,j_p} \}$ will be eliminated in the selection of a vertex corresponding to the decoded word.

Hence, $v_{i,j}$ can be eliminated from further considerations in soft decoding. □

Example 5.5.3 Consider the Example code for which the code tree is given in Figure 5.4.1 and the reduced tree is given in Figure 5.5.3.

If the received word $r_0r_1r_2$ is such that,
$d(0, r_0) = 0.03$,
$d(0, r_1) = 0.04$ and
$d(0, r_2) = 0.01$.
Then the path metric at vertex $v_{3,0}$ is, $d_{v_{3,0}} = 0.08$.
Now, if $d(2, r_0) = 1.5$.
And, if the computation in the other branches of the tree starts, then,
$d_{v_{2,2}} = 1.5 > d_{v_{3,0}} = 0.08$,
and further processing along this branch in the tree can be aborted and the tree can be reduced.

5.6 Considerations for Implementation of the Soft Decoder

This section considers implementation issues for using the reduced tree to perform soft decoding for BCM schemes. The reduced tree representation for the code words provides various trade offs in the implementation of the soft decoder.

The soft decoder basically consists of the following three components.

The program memory: The program memory stores the software program for the soft decoder. The code words represented as the tree with various reduction possibilities, is also present in the program memory. Besides, the data words corresponding to each vertex of the tree at level n, that is, the code table is also stored in the program memory. Hence, the size of the program memory required for soft decoding increases with the size of the code.

The data memory: The data memory stores all the Euclidean distances generated during the soft decoding of a received word. This consists of the Euclidean distances between the symbol of the received word and the channel symbols from the expanded channel signal constellation and the partial and final path metrics stored at the vertices of the tree. The data memory stores real numbers.

Processing element: The processing element executes the soft decoding software from the program memory and has to perform the following operations.

- Compute Euclidean distances between the received symbol and the symbols associated with the edges of the code tree.
- Read and write the Euclidean distances in the data memory.
- Add the Euclidean distances and obtain the path metrics.
- Compare the Euclidean distances to obtain the minimum Euclidean distance.

In a parallel soft decoder there will be a central processing element which controls the various other processing elements, working in parallel.

Data memory storage and the processing requirements are of prime concern in the design of a decoder. These issues are discussed in this section.

5.6.1 Reducing the Data Memory Storage

In a tree, back tracking is not required, as the data words are uniquely associated with the vertices of the tree at level n. Hence, as soon as the partial path metrics at all the vertices at a level i have been computed, the Euclidean distances between the received channel symbol and the edges of the tree, for all the symbols, above the level i can be deleted. This results in the reduction of the storage required for data memory.

Reduction of the code tree, reduces the number of vertices in a tree. During soft decoding a path metric has to be stored at each vertex of the tree. Hence, reduction of the tree reduces the data memory requirements of the decoder. So the various schemes described in the previous section for tree reduction should be used in the implementation of the soft decoder.

In the computation of the path metrics in the reduced tree, starting from the root vertex, as soon as the partial path metrics at the vertices at level i are found the partial path metrics for the vertices at level $i-1$ can be removed.

5.6.2 Exploiting Parallelism Provided by the Code Tree

A parallel soft decoder can be implemented at the cost of increased storage and complexity. In the decoding process, the level i' at which comparisons are to be made for deleting vertices from further consideration are known. The information about the subtrees at the vertices at level i', which are isomorphic with weights for a subset of vertices, of which only one is to survive, is also present. Also, the full received word of length n is known.

Parallel computations of the Euclidean distances of the received channel symbol, from the channel symbols of the expanded channel signal constellation, denoted by the edges of the tree, for each path from the root vertex to the vertices at level i' in the subset can be started. With this again in parallel, computations in the subtree can be started. If further sub-subtrees exist, their computations can also be initiated. As soon as all the metrics of the vertices in the subset at level i' have been obtained, computation of the minimum and deletion of other vertices can take place. As soon as the minimum is known and the computations in the subtree are over, to this minimum, the metric at the vertices of the subtree at level n can be added. Once these are known using Theorem 5.5.4, an upper bound can be found and the computations for all other subsets at level i' can proceed.

To further increase parallelism, computations for the vertices of all subsets at level i' and all the known subtree computations can be started simultaneously. If the storage of data during decoding is not a problem, then instead of the repeated calculations of the distances between the channel symbol denoted by the edge and the received symbol for a level, the matrix of Euclidean distances between the received symbols and the channel symbols can be computed only once and stored till the decoding of the word has been completed. In this way increase in parallelism is achieved at the cost of increase in the data memory store.

Example 5.6.1 Consider the Example code for which the code tree is given in Figure 5.4.1 and the reduced tree is given in Figure 5.5.3.

The storage and the comparisons required, for the soft decoding using the reduced tree

are given in Table 5.6.1. A temporary storage for 8 real numbers, and 7 comparisons are

Table 5.6.1: Comparison and storage required for the code in Example 5.6.1

Received symbols	Metric stored at the vertices	Total real no. store	Comparison of vertices	Total no. of compar's
r_1	$\{d_{v10}, d_{v11}, d_{v12}, d_{v13}\}$	4	—	—
r_2	$\{d_{v20}, d_{v21}, d_{v22}, d_{v23}, d_{v24}, d_{v25}, d_{v26}, d_{v27}\}$	8	— —	— —
			$(d_{v20} \leq d_{v23})$ $(d_{v21} \leq d_{v22})$ $(d_{v24} \leq d_{v26})$ $(d_{v25} \leq d_{v27})$	4
	$\{d_{v20}, d_{v21}, d_{v24}, d_{v25}\}$	4		
r_3	$\{d_{v30}, d_{v31}, d_{v34}, d_{v35}\}$	4	Find minimum	3

required.

For an parallel implementation the following computations can be performed in parallel, $d(0, r_1) + d(0, r_2)$, $d(1, r_1) + d(2, r_2)$ and $d(0, r_3)$.

Then compare if $d_{v2,0} \leq d_{v2,3}$ and find $d_{v3,0}$.

Similar computations in the other three isomorphic subtrees can also be performed in parallel.

5.7 Algorithm for Soft Decoding Using a Reduced Tree

In this section, the scheme for obtaining a reduced tree and using it for soft decoding of a block code, used for a BCM scheme, is summarized in the following algorithm.

(1) For a block code to be used for a BCM scheme, obtain the code tree from the code words.

(2) Based on Theorem 5.5.2 identify vertices with subtrees isomorphic with weights and reduce the code tree. In fact, the reduction can take place simultaneously with the generation of the tree.

(3) On the basis of Theorem 5.5.3, obtain codes equivalent to the block code in use. Repeat steps (1) and (2), for the equivalent codes. Obtain an equivalent code which results in the reduction of the maximum vertices of the tree.

(4) Based on the equivalent code selected for use in the soft decoder, either perform similar permutation on the received word or modify the encoder as discussed in Section 3.8.3.

(5) Assign the data words with the vertices of this tree at the level n.

(6) Depending on the application, select proper number of processors and data memory storage for the soft decoder.

(7) Using the processors and the memory with the reduced tree, obtain the path metrics at the vertices starting from level 1 to level n.

(8) When the path metric at each vertex is computed, compare it with the bounds obtained based on Theorem 5.5.4. For GU codes the bound based on Theorem 5.5.5 also exists.

(9) At level n, obtain a vertex with the minimum metric.

(10) The data word corresponding to this vertex is the soft decoded data word.

5.8 Examples

In this section the reduced tree corresponding to the codes given in the examples[6] of Chapters 3 and 4 are given.

Example 5.8.1 The reduced tree for the code in Example 3.3.1 is shown in Figure 5.8.1.

Example 5.8.2 The reduced tree for the code {0000, 1120, 1102, 1311, 3111, 0231. 2031, 0213, 2013, 2200, 0022, 3320, 3302, 1333, 3133, 2222}, which is a code equivalent to the code in Example 3.10.1 is shown in Figure 5.8.2.

Example 5.8.3 The reduced tree for the code in Example 3.10.2 is shown in Figure 5.8.3.

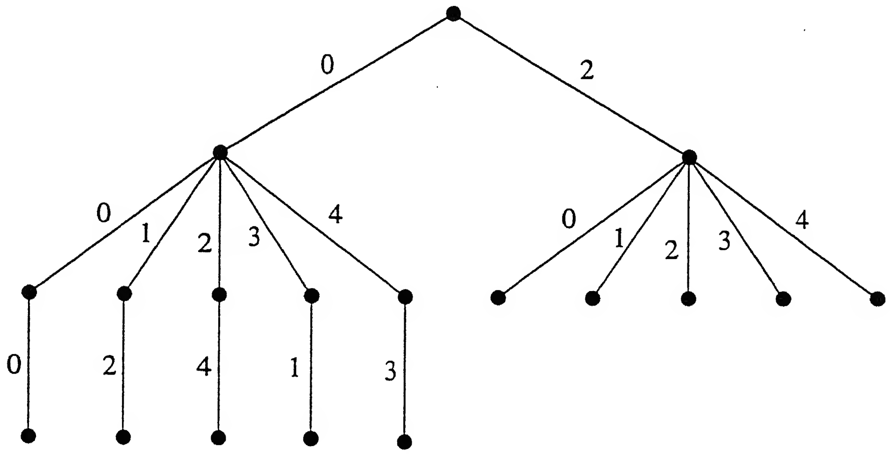

Figure 5.8.1: Reduced code tree for Example 5.8.1

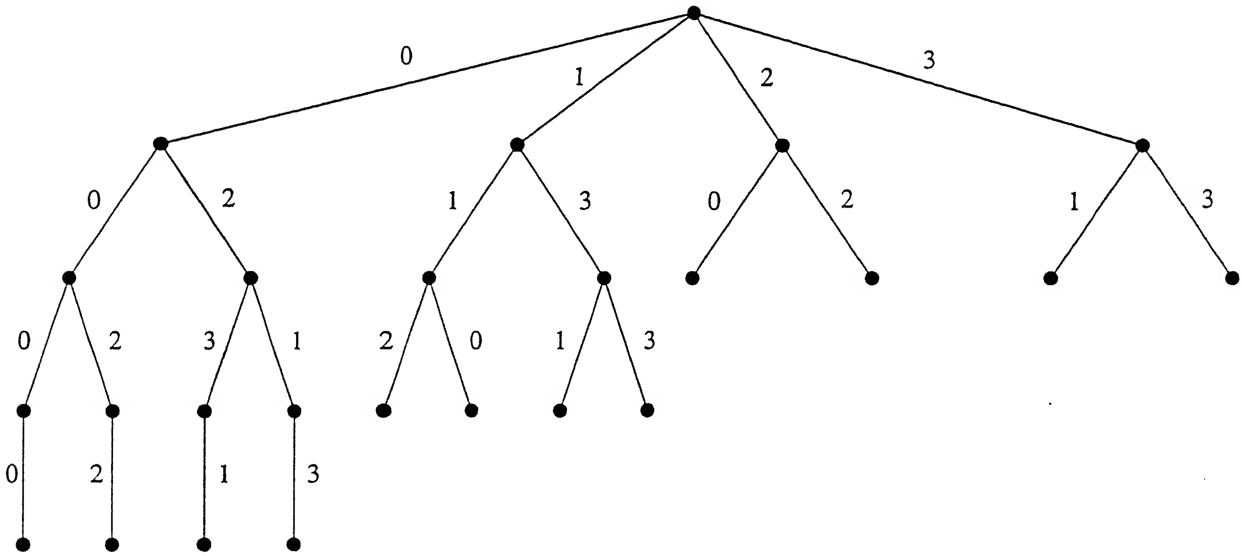

Figure 5.8.2: Reduced code tree for Example 5.8.2

Example 5.8.4 The reduced tree for the code in Example 4.8.1 is shown in Figure 5.8.4.

Example 5.8.5 The reduced tree for the code in Example 4.8.2 is shown in Figure 5.8.5.

[6]In all the examples discussed here for comparison, vertices with lower j are assumed to have minimum Euclidean distance at a level.

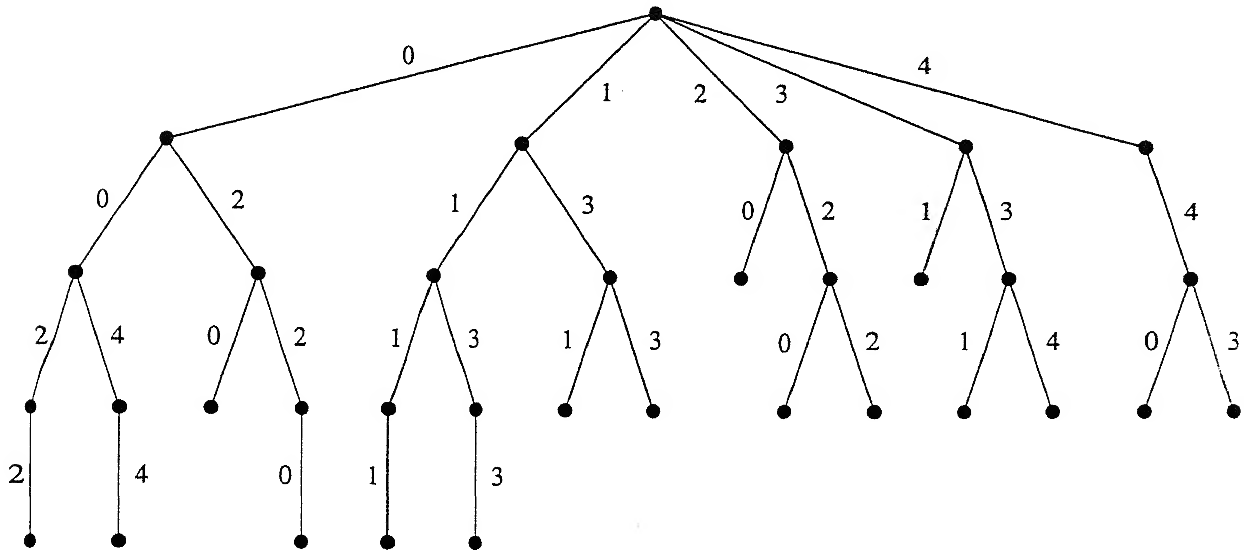

Figure 5.8.5: Reduced code tree for Example 5.8.5

mentation. Techniques for obtaining a reduced tree from the code tree and considerations regarding the implementation of the soft decoder are discussed. The scheme does not require back tracking and uses the full block code of length n simultaneously for decoding.

Chapter 6

Obtaining An Optimal Minimal Trellis for General Block Codes

6.1 Introduction

This chapter proposes the use of an optimal minimal trellis for soft decoding of general block codes. A scheme of using equivalent codes, for obtaining the optimal minimal proper trellis for the soft decoding of general block codes, is developed. The block codes obtained by the structured distance approach in Chapters 3 and 4 need not have linear, cyclic, group, GU and rectangular structure on the code words. For such general codes an optimal minimal proper trellis can be obtained using the presented scheme. A condition for obtaining the optimal minimal improper trellis, and hence the optimal minimal trellis, for general block codes of block length 3, is derived.

The optimal minimal proper trellis for a general code is obtained from the code tree of an equivalent code, discussed in Section 3.3.2. Properties of the trellis as implied from the nature of the code tree are also discussed.

The chapter begins with a brief overview of some essential terms and relevant references. A comparison of the use of code tree and trellis for the representation of block codes is given. Issues relating to the minimal trellis for block codes are discussed. Based on the code trees for equivalent codes obtained in Section 5.5.2, a scheme is presented for obtaining the optimal minimal proper trellis for a general block code. A condition for obtaining the optimal minimal trellis for codes with block length $n = 3$, is stated. An algorithm summarizes

the procedure for obtaining the trellis. Optimal minimal trellises for the example codes of Chapter 3 and 4, are obtained. Finally, the chapter ends with some concluding remarks.

6.2 Background and Preliminaries

Traditionally, for coded modulation a trellis[1] representation of the code is used in the soft decoder. Block codes and BCM schemes also usually use a trellis for soft decoding.

For the convolutional code a trellis for the code is identical in every stage. It is a common practice. to represent a convolutional code by drawing the trellis for a single stage. For a linear cyclic block code also the trellis is periodic [90], like the convolutional code, except for the fact that, there is a single initial and final state in the trellis. But, for a general (non-linear) codes the trellis need not have any regular structure.

For linear block codes the trellis is obtained using the generator matrix for the code [90] and using this soft decoding is performed. Matis and Modestino [60], have given a reduced search soft decision trellis decoding scheme for linear block codes. The scheme presented by Wolf [90] has been used by Isaksson and Zetterberg [42] for the soft decoding of linear block codes for BCM. The minimal trellis was introduced by Forney [26]. Muder [61] has discussed the minimal proper and improper trellises for block codes and has presented a scheme for obtaining an minimal proper trellis for a block code from a tree. Some important results regarding the trellis structure for block codes have been presented by Kschischang and Sorokine [51]. Lower bounds on the trellis complexity of block codes have been stated by Lafourcade and Vardy [52]. Vardy and Kschischang [82] show that the minimal trellis can be obtained for a class of codes known as rectangular codes which include all group codes and some non-linear codes. The trellis structure of maximal fixed cost codes has been studied by Kschischang [49]. An efficient algorithm for constructing minimal trellises for codes over finite abelian groups has been presented by Vazirani, Saran and Sundar Rajan [83].

Forney [26] showed that the state complexity of the minimal trellis for a linear code depends on the order of its bit positions. Kasami et al. [47, 48] have found optimum bit ordering for reducing the state complexity for trellises of binary linear codes. Schuurman [71]

[1] In fact, it is due to the use of trellises for the representation of the codes that schemes using convolutional codes got the name **trellis coded modulation [TCM]**.

has obtained complexity bounds for binary linear codes. Horn and Kschischang [40] have commented on the intractability of the problem of permuting a block code to minimize the trellis complexity. Compared to the existing literature, this thesis concentrates on using general block codes of short length with BCM schemes. The proposed scheme is different since an equivalent code is used to obtain an optimal minimal proper trellis for a general block code which might not be even a rectangular code. Optimal minimal improper trellises are also obtained for $n = 3$.

After obtaining the trellis, the Viterbi algorithm [24, 84] is used with the trellis representation of the code words for soft decoding. Gulak and Shwedyk [36], and, Gulak and Kailath [35] have discussed VLSI implementation schemes for the Viterbi algorithm.

6.3 The Tree and the Trellis

The code tree and its use for the soft decoding of block codes has been discussed in details in Chapter 5. For the sake of comparison, the following characteristics of the tree as a representational scheme for the code words can be noted.

(1) In a code tree there is only one edge entering any vertex of the tree, except for the root vertex.

(2) A single path metric, computed at the edge entering a vertex, is stored at each vertex of the code tree.

(3) The number of vertices at level n is equal to $|C|$, the number of code words for the block code. The vertices at level n are in one-to-one correspondence with the data words.

(4) The soft decoding computations, in the various subtrees at the distinct vertices, at any level can proceed independently.

(5) Soft decoding with the code tree uses the full code word of block length n simultaneously.

(6) After a vertex at the final level corresponding to the received word is obtained, no backtracking is required.

(7) Block codes have a code tree unique up to isomorphisms.

A trellis, like a tree, is also a structure for representing the code words for the purpose of soft decoding.

Definition 6.3.1 *A trellis of length n is an edge-labeled directed graph, such that the set of vertices can be decomposed into a union of disjoint subsets,*

$$(\mathrm{V}_0 = \{ v_{0,0} \}) \cup \mathrm{V}_1 \cup \mathrm{V}_2 \ldots \cup (\{ v_{n,0} \} = \mathrm{V}_n),$$

such that every edge that begins at a vertex at level i, ends at a vertex at level $i + 1$. $0 \leq i \leq n - 1$ and $v_{0,0}$ is the initial vertex and $v_{n,0}$ is the final vertex.

The trellis has the following characteristic properties related with representation of code words for soft decoding.

(1) In a trellis, multiple edges can enter any vertex except $v_{0,0}$.

(2) A branch metric corresponding to each incoming edge has to be partially stored at each vertex of the trellis.

(3) The trellis has a single vertex $v_{n,0}$ at level n.

(4) Due to the strong interconnectivity between the vertices of a trellis, computations inside various sections of the trellis cannot proceed independently.

(5) Soft decoding using the trellis proceeds stage by stage in a sequential manner.

(6) Once the computations at the final vertex are over, backtracking is required to obtain the decoded word.

(7) Various trellises exist for a block code.

Example 6.3.1 Consider the block code for the (2-PSK, 4-PSK, 3, 8, $\sqrt{6}$) BCM scheme for which a code tree has been illustrated in Example 5.4.1.

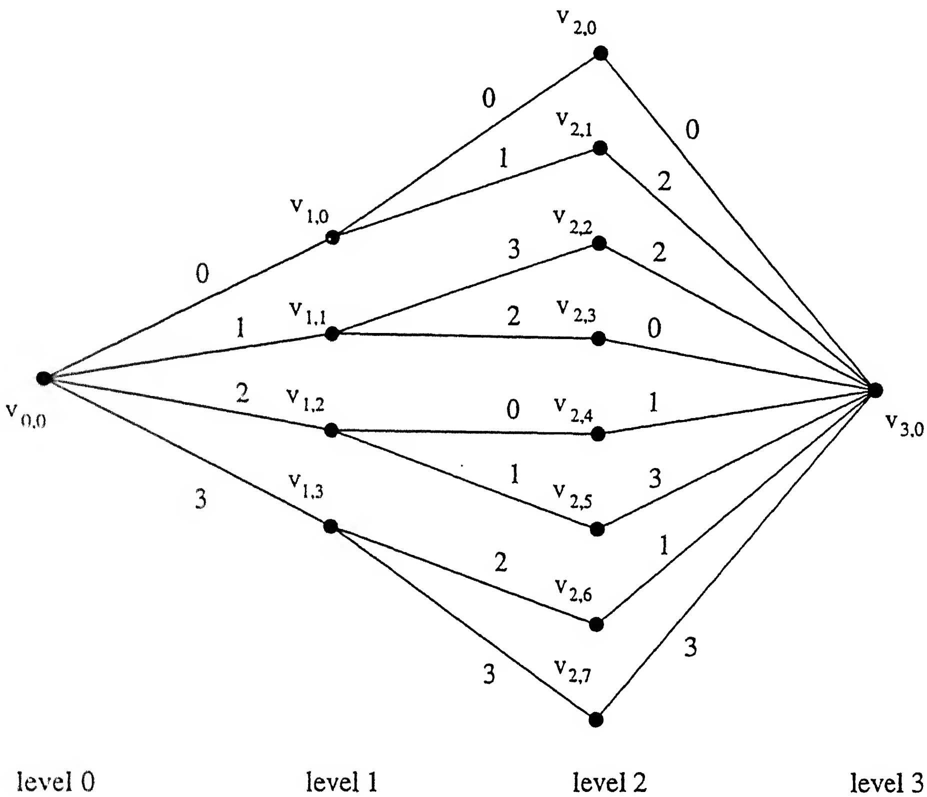

Figure 6.3.1: Trellis representation for the code in Example 6.3.1

The code words are $C = \{\, 000,\ 012,\ 132,\ 120,\ 321,\ 201,\ 213,\ 333 \,\}$.
A trellis for this code is given in Figure 6.3.1. This trellis has a single vertex at level 0—$v_{0,0}$,
4 vertices at level 1—$v_{1,0}$, $v_{1,1}$, $v_{1,2}$ and $v_{1,3}$,
8 vertices at level 2—$v_{2,0}$, $v_{2,1}$, $v_{2,2}$, $v_{2,3}$, $v_{2,4}$, $v_{2,5}$, $v_{2,6}$ and $v_{2,7}$, and
a single final vertex at level 3—$v_{3,0}$.
The total number of vertices in this trellis is 14.
Another trellis for the same code is shown in Figure 6.3.2. This trellis has a single vertex at level 0,
4 vertices at level 1,
4 vertices at level 2 and
a single vertex at level 3.
The total number of vertices in this trellis is 10.

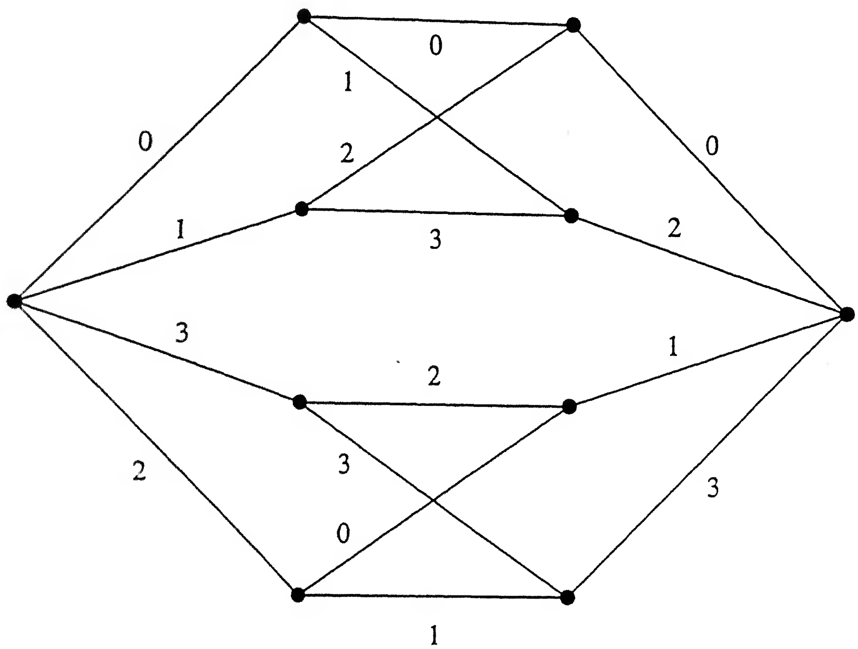

Figure 6.3.2: Another trellis representation for the code in Example 6.3.1

A trellis found for a block code can be used for soft decoding the received word with the Viterbi algorithm. The Viterbi algorithm, which is based on the principles of dynamic programming, is used to find the path in the trellis, that is, the code word with the minimum path metric (Euclidean Distance) from the received word. The soft decoding is performed by sequentially moving through the trellis stage by stage [15]. The procedure for soft decoding using the Viterbi algorithm with a trellis is briefly summarized.

(1) Starting from the initial vertex at level 0, take the first received symbol of a received word and find its Euclidean distance with the labels at the edges of the trellis which start from vertices at level 0 and end at level 1. These Euclidean distances associated with each edge of the trellis are known as the branch metrics.

(2) For all the branches terminating at a vertex of the trellis, the minimum branch metric is stored at the vertex at level 1 and is known as the path metric.

(3) For vertices at level i, $2 \leq i \leq n-1$, to the path metrics at the vertices of level i, the branch metrics for the edges from level i to level $i+1$ are added, and the minimum

metric is stored as the path metric at the vertices at level $i+1$.

(4) This procedure is repeated stage by stage, till the path metric at the final vertex at level n has been found.

(5) The soft decoded word is obtained by backtracking and traversing from the final vertex to the initial vertex, in such a manner that at each stage a vertex with the minimum path metric is selected.

To perform soft decoding when the Viterbi algorithm is used with a trellis, the following observations can be stated.

- At each vertex of the trellis which has more than one incoming edge, comparisons have to be made to find the path metric.
- At each vertex of the trellis which has more than one incoming edge, the distance has to be stored temporarily in the data memory for each edge before the path metric at the vertex can be computed.
- In the trellis the path metric at all the vertices, have to be stored in the data memory.
- During backtracking comparisons have to be made at each level in the trellis.

6.4 The Optimal Minimal Trellis

In a trellis based soft decoder for block codes, it is advisable to utilize a trellis which is most efficient for use with the Viterbi algorithm. As considered in Section 5.6.1 for the case of the reduced tree, here too, the data memory storage is the issue of prime concern. The data memory storage required is directly proportional to the number of vertices in the trellis. Hence, the minimization of the number of vertices in the trellis at a stage, is the primary criteria for obtaining an efficient trellis representation. This is the basic motivation for the use of the *minimal trellis* for a block code. In this thesis, a different term the *optimal minimal trellis*, is introduced, for referring to the minimal trellis obtained from considerations of code equivalence.

In the literature various authors have given different definitions of the minimal trellis for a code. For example,

- Muder [61] has defined a minimal trellis as a trellis having the number of vertices at all the levels less than the number of vertices at the corresponding levels of any other trellis. This is the definition for minimal trellis used in the thesis.
- Lafourcade [52] has used a trellis with the smallest possible maximum vertex count, where the maximum is taken over the vertex sets at different times as a minimal trellis.
- Kschischang [51] has suggested the use of a trellis having the minimum number of vertices in total as the minimal trellis.

Definition 6.4.1 *A trellis with vertices* V_i *such that the cardinality of* $\mathrm{V}_i \leq$ *cardinality of* V_i', $\forall i$, *where* V_i *and* V_i' *are the vertices of trellises for the same code is known as the minimal trellis [61].*

Definition 6.4.2 *A trellis for a block code of length n, is proper [61],*
if it has a single initial vertex $v_{0,0}$,
every vertex of the trellis lies on some path of length n starting from $v_{0,0}$, *and,*
no two different paths from $v_{0,0}$ *of some length i correspond to the same i-tuple, that is, no two edges emanating from a vertex have the same label on the edges.*

Definition 6.4.3 *A trellis which is not a proper trellis is known as an improper trellis*

Definition 6.4.4 *The minimal trellis for a block code which is proper is known as the minimal proper trellis.*

Definition 6.4.5 *The minimal trellis for a block code which is improper is known as the minimal improper trellis.*

The following interesting facts regarding the nature of the minimal trellis for general block codes can be stated.

FACT 1: Every general block code has a minimal proper trellis, and any two minimal proper trellises for the same code are isomorphic [61, Theorem 1].

FACT 2: For a general code the minimal proper trellis might not be the minimal trellis [61].

FACT 3: A minimal trellis for a general (non-linear) code may not be unique [51].

FACT 4: The determination of the minimal trellis for a general (non-linear) code appears to be computationally infeasible in general and is NP-complete [51].

FACT 5: For obtaining a minimal trellis for a general code, a minimal vertex assignment at one time index may not be compatible with a minimal assignment at some other time index [51].

Example 6.4.1 Of the two trellises obtained in Example 6.3.1, the trellis shown in Figure 6.3.2 is a minimal proper trellis and for the block code it is also the minimal trellis.

Equivalent codes can be used for soft decoding using the trellis [26, 40, 47, 48]. As stated in the Lemma 5.5.1, equivalent codes have a different trellis representation. The use of an equivalent code will be advantageous if it gives a more efficient trellis than the minimal trellis, called the **optimal minimal trellis** for a block code.

Definition 6.4.6 *A minimal trellis, with vertices* V_i *such that the cardinality of* $\mathrm{V}_i \leq$ *cardinality of* $\mathrm{V}_i{}'$, $\forall i$, *where* V_i *and* $\mathrm{V}_i{}'$ *are the vertices of minimal trellises for the equivalent codes, is known as the optimal minimal trellis.*

For a general block code the optimal minimal trellis might not be unique, but a set of optimal minimal trellises can exist. The optimal minimal trellis might be proper or improper.

Definition 6.4.7 *The optimal minimal trellis for a block code which is proper is known as the optimal minimal proper trellis.*

Definition 6.4.8 *The optimal minimal trellis for a block code which is improper is known as the optimal minimal improper trellis.*

If a trellis based on an equivalent code has to be used for the soft decoding of the general block code, the following considerations have to be introduced in the soft decoding scheme.

(1) When more than one optimal minimal trellis exists, any of the trellises will give same performance for soft decoding.

(2) The code table at the receiver has to be modified and should use the equivalent code or else the encoder has to changed to use the equivalent code.

(3) The soft decoding can start only after a block code is received, if the encoder is not modified after obtaining the optimal minimal trellis. Hence a buffer is required to store the received word.

(4) The permutation of the co-ordinate positions made on the code words has to be applied to the received word for soft decoding, if the encoder is not modified.

Example 6.4.2 Of the two trellises obtained in Example 6.3.1, the trellis shown in Figure 6.3.2 is also the optimal minimal proper trellis and the optimal minimal trellis for the block code.

6.5 Motivating Factors

(1) For the purpose of soft decoding an equivalent code can be used to obtain an optimal minimal trellis. For a general code this optimal minimal trellis may be proper or improper. It might not be even a unique optimal minimal trellis, but there might be a set of optimal minimal trellises giving similar performance (criteria decided for soft decoding). For a general block code which might not be linear, group, rectangular, etc. a scheme to obtain the optimal minimal proper trellis is required.

(2) If the optimal minimal trellis for a general code is improper. At least, for certain cases it is interesting to see if the optimal minimal improper trellis can be found.

6.6 Obtaining Optimal Minimal Proper Trellises for General Block Codes

This section considers the problem of obtaining the optimal minimal proper trellis for general block codes from the code tree discussed in Chapter 5. Note that, the optimal minimal proper trellis might not be the optimal minimal trellis for a general block code.

A code tree has only one edge entering a vertex from the previous level, but in a proper trellis, at most n' edges can enter a vertex, from various vertices of the previous level. A minimal proper trellis can be obtained from the code tree by merging together the vertices of the tree at a level.

Definition 6.6.1 *Let, $IV_{i,j}$ denote the edge of the tree into the vertex $v_{i,j}$, for $1 \leq i \leq n$ and $\forall$ j.*

Definition 6.6.2 *Let, $OV_{i,j,q}$, $0 \leq i \leq n-1$ and $\forall$ j, denote the edges of the tree out of the vertex $v_{i,j}$, where q is such that $0 \leq q \leq n' - 1$. n' is the number of signals in the expanded channel signal constellation used for the block code.*

Definition 6.6.3 *Let, $\mathbf{IV}_{i,k,p}$, for $1 \leq i \leq n$ and $\forall$ k, denote the edges of the trellis into a vertex $\mathbf{V}_{i,k}$, where $0 \leq p \leq n' - 1$.*

Definition 6.6.4 *Let, $\mathbf{OV}_{i,k,q}$ $0 \leq i \leq n-1$ and $\forall$ k, denote the edges of the trellis out of the vertex $\mathbf{V}_{i,k}$, where $0 \leq p \leq n' - 1$.*

To obtain the minimal proper trellis from the code tree, vertices are found at a level having subtrees isomorphic with weights. These vertices are then merged, instead of deleting them as discussed in Section 5.5.

Definition 6.6.5 *Let, v_{i,j_1} and v_{i,j_2} be two vertices of the tree at level i, which are merged to obtain a vertex $\mathbf{V}_{i,j}$ of the minimal proper trellis. Then, the union of the set of incoming edges is the set of the incoming edges for the resulting vertex of the trellis. And, the union of the set of outgoing edges is the set of the outgoing edges for the resulting vertex of the trellis. That is,*

$$\mathbf{IV}_{i,j,p} = IV_{i,j_1} \cup IV_{i,j_2} \text{ and}$$

$$\mathrm{OV}_{i,j,p} = OV_{i,j_1,q} \cup OV_{i,j_2,q} \;\forall q.$$

Theorem 6.6.1 *The code tree, in which all the vertices in the partition set $\mathbf{V}_{i,k}$, $1 \leq i \leq n$ and $\forall$ k, are merged into a single vertex, is equivalent to the full code tree for the purpose of soft decoding.*

Proof: At a level i, $\exists$ k', (let $k = k'$) such that $0 \leq k' \leq l_i - 1$ and cardinality of $\mathbf{V}_{i,k'} > 1$.

CASE 1:

If the cardinality of $\mathbf{V}_{i,k} = 1$, for some k, then no merging is required and hence the theorem holds good.

CASE 2:

If cardinality of $\mathbf{V}_{i,k} > 1$,

then let cardinality of $\mathbf{V}_{i,k} = m_{i,k}$, where, $m_{i,k} > 1$.

From Definition 5.5.3,

$$\mathbf{V}_{i,k} \subset \{\, v_{i,j} \,\} = \{\, v_{i,j_0},\ v_{i,j_1},\ \ldots\ v_{i,j_{m_{i,k}-1}} \,\}.$$

From Definition 5.5.3 and Definition 5.5.4 it follows that,

$$\mathbf{T}_{v_{i,j_0}} \overset{\mathrm{W}}{\cong} \mathbf{T}_{v_{i,j_1}} \overset{\mathrm{W}}{\cong} \ldots \overset{\mathrm{W}}{\cong} \mathbf{T}_{v_{i,j_{m_{i,k}-1}}}.$$

Hence, $\{\, v_{n,j_{0p}} \,\} \overset{\mathrm{W}}{\cong} \{\, v_{n,j_{1p}} \,\} \overset{\mathrm{W}}{\cong} \ldots \overset{\mathrm{W}}{\cong} \{\, v_{n,j_{m_{i,k}-1p}} \}$, where, $1 \leq p \leq p'$, and p' = the number of vertices of the subtree at level n of the tree.

Hence, $\{\, d(v_{n,j_{0p}} - v_{i,j_0}) \,\} = \{\, d(v_{n,j_{1p}} - v_{i,j_1}) \,\} = \ldots = \{\, d(v_{n,j_{m_{i,k}-1p}} - v_{i,j_{m_{i,k}-1}}) \,\}$, $\forall$ p.

Here, $d(v_y - v_x)$, denotes the sum of path metrics for the partial path from the vertex at level x to the vertex at level y, assuming that $dv_x = 0$. Where, $dv_{i,j}$ denotes the path metric along a path through vertex $v_{i,j}$, stored at the vertex $v_{i,j}$, $0 \leq i \leq n$, of the tree.

So a single subtree at level i for $\mathbf{V}_{i,k}$ is sufficient for the purpose of soft decoding.

So all the vertices of the tree in $\mathbf{V}_{i,k}$ are merged for the purpose of soft decoding.

From Definition 6.6.1 the union of the edges entering the vertex is formed and a path metric which is the minimum branch metric will be stored at the vertex. So the decision to be made at level n, for vertices at level n having a vertex in $\mathbf{V}_{i,k}$ as the parent vertex, is made at level i itself.

Hence, the code tree in which the vertices in the partition set $\mathbf{V}_{i,k}$ are merged into a single vertex, is equivalent to the full code tree for the purpose of soft decoding. $\square$

Theorem 6.6.2 *In the code tree, by merging the vertices at level i, $\forall\, i$ such that $1 \leq i \leq n-1$. according to Theorem 6.6.1, and by merging all the vertices at level n, the minimal proper trellis for the code is obtained, as in [61, Proposition 2].*

Proof: The code tree is an edge labeled directed graph, in which an edge begins at a vertex at level i and ends at a vertex at level $i+1$. At level 0, it has a single vertex $\mathbf{V}_0 = \{\, v_{0,0} \,\}$. For level $i = 1$ to $n-1$, by merging the vertices at each level according to Definition 6.6.1 and Definition 6.6.2.

$$\mathbf{V}_i = \{\, \mathbf{V}_{i,k_1},\ \mathbf{V}_{i,k_2},\ \ldots\ \mathbf{V}_{i,k_q} \,\}, \text{ for some } q.$$

By merging all the vertices at the last level, $\mathbf{V}_n = \{\mathbf{V}_{n,0}\}$, is obtained.

$$\cap_{\forall i} \mathbf{V}_i = \emptyset,$$

since each i corresponds to a separate level of the code tree.
Hence, from Definition 6.3.1, by merging vertices of the code tree according to Theorem 6.6.1 and by merging all vertices at level n, we obtain a trellis.
For a set of code words, the code tree is unique, if isomorphisms are disregarded.
From Definition 5.5.3 and Definition 5.5.4 it follows that, the partitioning at a level and hence merging is a exhaustive process, resulting in minimum vertices at a level. No other minimization is possible, for a code tree, without affecting the soft decoding process. Hence, from Definition 6.4.1, the trellis obtained is a minimal trellis.
Since the conditions of Definition 6.4.2 are satisfied, the trellis obtained is the minimal proper trellis for the code. □

As a consequence of Theorem 6.6.1 and Theorem 6.6.2 the minimal proper trellis can be used for soft decoding of a block code, instead of the code tree. It can be observed that,

(The number of states in a minimal proper trellis at level 0 and level n) $= 1$,

$1 \leq$ (The number of states in a minimal proper trellis at level 1) $\leq n'$,

$1 \leq$ (The number of states in a minimal proper trellis at level i, $1 < i \leq n-1$) $\leq n'^i$.

Example 6.6.1 Consider the code tree obtained in Example 5.4.1. The code words are $C = \{\, 000. 012, 132, 120, 321, 201, 213, 333 \,\}$.

At level 0, only the root vertex is present and no merging is possible.

At level 1, also the subtrees at the 4 vertices of the code tree are not isomorphic with weight, hence no merging is possible.

At level 2, the code tree has 8 vertices and,

$T_{v_{2,0}} \overset{w}{\cong} T_{v_{2,3}}$,

$T_{v_{2,1}} \overset{w}{\cong} T_{v_{2,2}}$,

$T_{v_{2,4}} \overset{w}{\cong} T_{v_{2,6}}$ and

$T_{v_{2,5}} \overset{w}{\cong} T_{v_{2,7}}$.

Hence, merging of the vertices is possible at this level.

At $V_{2,0}$:

$IV_{2,0,0} = 0$, $IV_{2,0,1} = 2$, and $OV_{2,0,0} = 0$.

At $V_{2,1}$:

$IV_{2,1,0} = 1$, $IV_{2,1,1} = 3$, and $OV_{2,1,0} = 2$.

At $V_{2,2}$:

$IV_{2,2,0} = 0$, $IV_{2,2,1} = 2$, and $OV_{2,2,0} = 1$.

At $V_{2,3}$:

$IV_{2,3,0} = 1$, $IV_{2,3,1} = 3$, and $OV_{2,3,0} = 3$.

At level 3 all the four vertices of the code tree $v_{3,0}$, $v_{3,1}$, $v_{3,4}$ and $v_{3,5}$ are merged to obtain the final vertex of the trellis.

$IV_{3,0,0} = 0$, $IV_{3,0,1} = 2$, $IV_{3,0,2} = 1$ and $IV_{3,0,3} = 3$. The minimal proper trellis is shown in Figure 6.6.1.

Theorem 6.6.3 *The minimal proper trellises for equivalent codes need not be identical.*

Proof: From Theorem 5.5.3 and Lemma 5.5.1, it follows that the code tree and the reduced code tree for equivalent codes are different.

From Theorem 6.6.2, it follows that the minimal proper trellis is obtained from the code tree by merging the vertices of the code tree.

Hence the minimal proper trellises for equivalent codes can be different.

Hence, the minimal proper trellises for equivalent codes need not be identical. □

Definition 6.6.6 *Let $\tilde{T}$, be the set of the minimal proper trellises for all the equivalent codes of a block code used with a BCM scheme.*

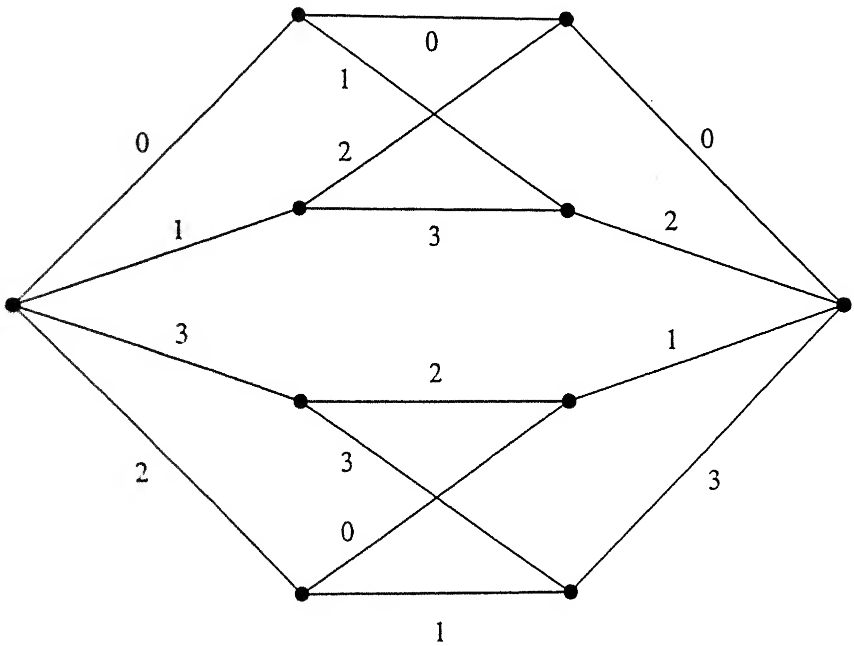

Figure 6.6.1: Minimal proper trellis for the code in Example 6.6.1

Definition 6.6.7 *A trellis, such that, it is $\in \tilde{\mathrm{T}}$ and satisfies the Definition 6.4.6 and 6.4.7 can be used as the optimal minimal proper trellis. Based on Theorem 6.6.3, this may not be unique for a block code.*

The scheme for obtaining the optimal minimal proper trellis for a block code is summarized in the following section.

Example 6.6.2 Consider the Example code C_0 for a BCM scheme, discussed in Example 5.5.2.
$C_0 = \{ 000, 011, 020, 021, 102, 110, 112, 121, 122\ 200\ 201, 202 \}$.

The minimal proper trellis for this code is shown in Figure 6.6.2. The number of vertices at each level of the minimal proper trellis are tabulated in Table 6.6.1. Consider the equivalent code C_1 obtained by permutation of the co-ordinate positions of C_0.
$C_1 = \{ 000, 011, 002, 012, 120, 101, 121, 112, 122\ 200\ 210, 220 \}$.
The minimal proper trellis for this code is shown in Figure 6.6.3.
For this code the number of vertices at each level of the minimal proper trellis are tabulated

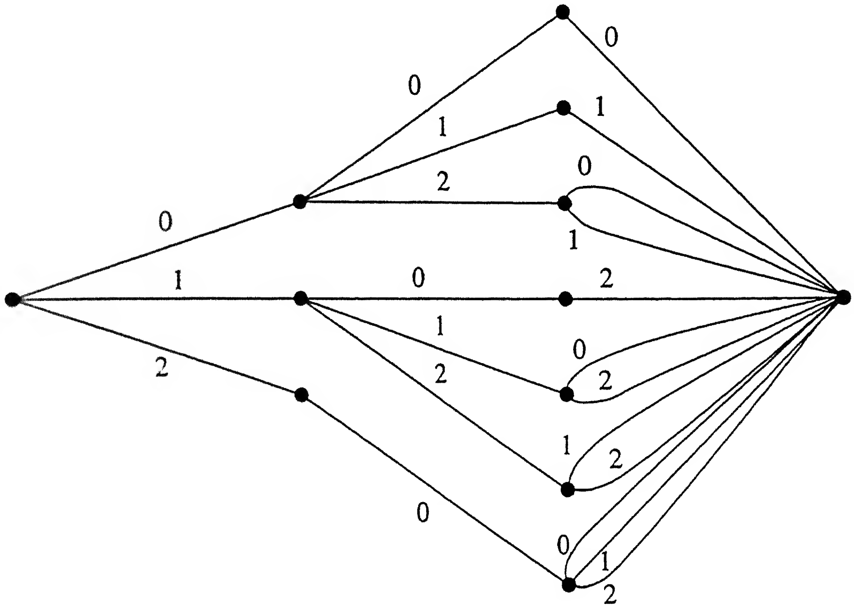

Figure 6.6.2: Minimal proper trellis for C_0 in Example 6.6.2

Table 6.6.1: Number of vertices in the minimal proper trellis of C_0 in Example 6.6.2

Level	No. of vertices in the minimal proper trellis
0	1
1	3
2	7
3	1

in Table 6.6.2. Consider the equivalent code C_2 obtained by permutation of the co-ordinate positions of C_0.
$C_2 = \{000. 110, 200, 210, 021, 101, 121, 211, 221\ 002\ 012, 022\}$.
The minimal proper trellis for this code is shown in Figure 6.6.4. For this code the number of vertices at each level of the minimal proper trellis are tabulated in Table 6.6.3. Consider the equivalent code C_3 obtained by permutation of the co-ordinate positions of C_0.
$C_3 = \{000, 101, 200, 201, 012, 110, 112, 211, 212\ 020\ 021, 022\}$.
The minimal proper trellis for this code is shown in Figure 6.6.5.

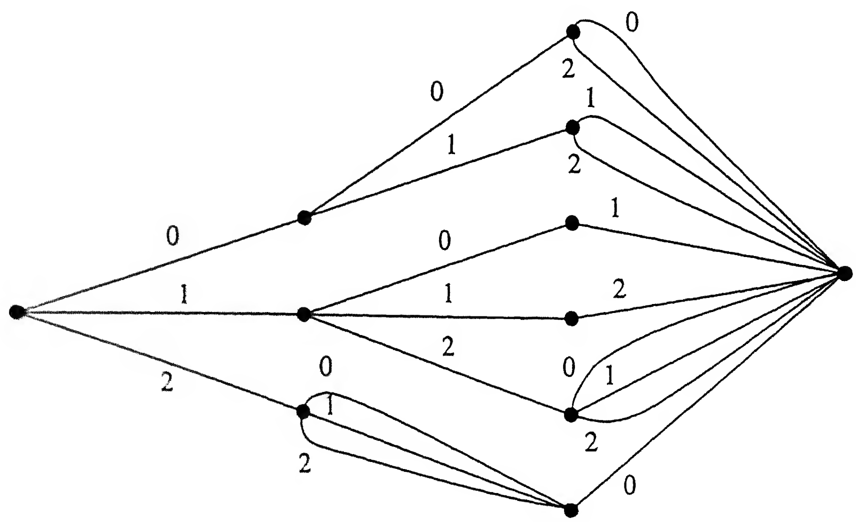

Figure 6.6.3: Minimal proper trellis for C_1 in Example 6.6.2

Table 6.6.2: Number of vertices in the minimal proper trellis of C_1 in Example 6.6.2

Level	No. of vertices in the minimal proper trellis
0	1
1	3
2	6
3	1

Table 6.6.3: Number of vertices in the minimal proper trellis of C_2 in Example 6.6.2

Level	No. of vertices in the minimal proper trellis
0	1
1	3
2	6
3	1

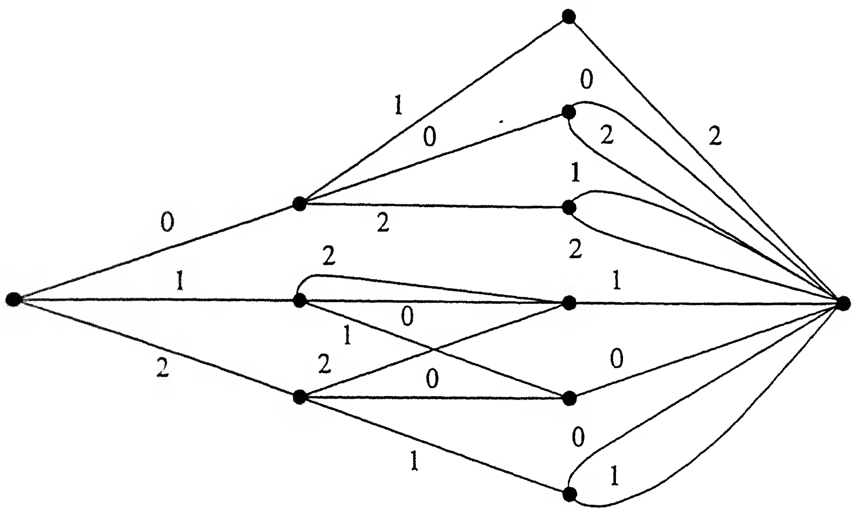

Figure 6.6.4: Minimal proper trellis for C_2 in Example 6.6.2

For this code the number of vertices at each level of the minimal proper trellis are tabulated

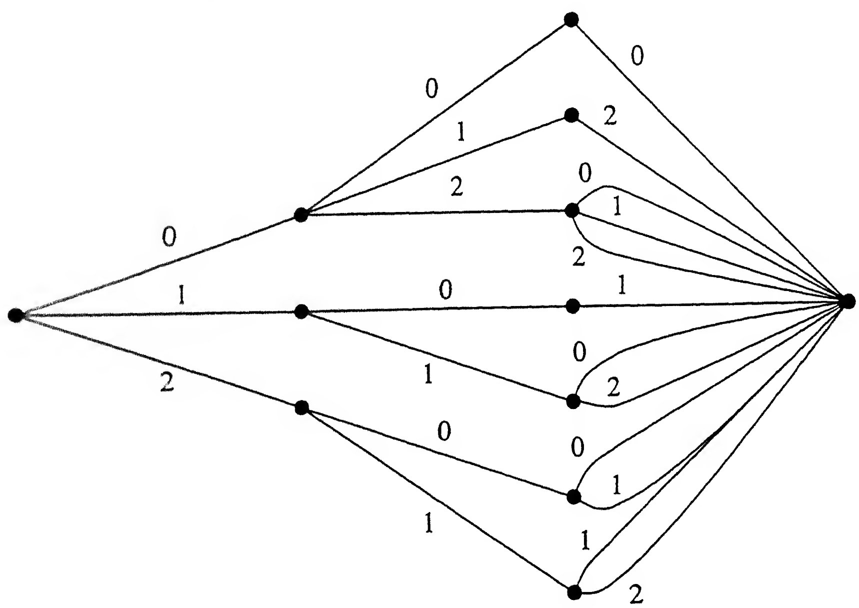

Figure 6.6.5: Minimal proper trellis for C_3 in Example 6.6.2

in Table 6.6.4. Consider the equivalent code C_4 obtained by permutation of the co-ordinate

Table 6.6.4: Number of vertices in the minimal proper trellis of C_3 in Example 6.6.2

Level	No. of vertices in the minimal proper trellis
0	1
1	3
2	7
3	1

positions of C_0.
$C_4 = \{ 000, 101, 002, 102, 210, 011, 211, 112, 212\ 020\ 120, 220 \}$.
The minimal proper trellis for this code is shown in Figure 6.6.6. For this code the number

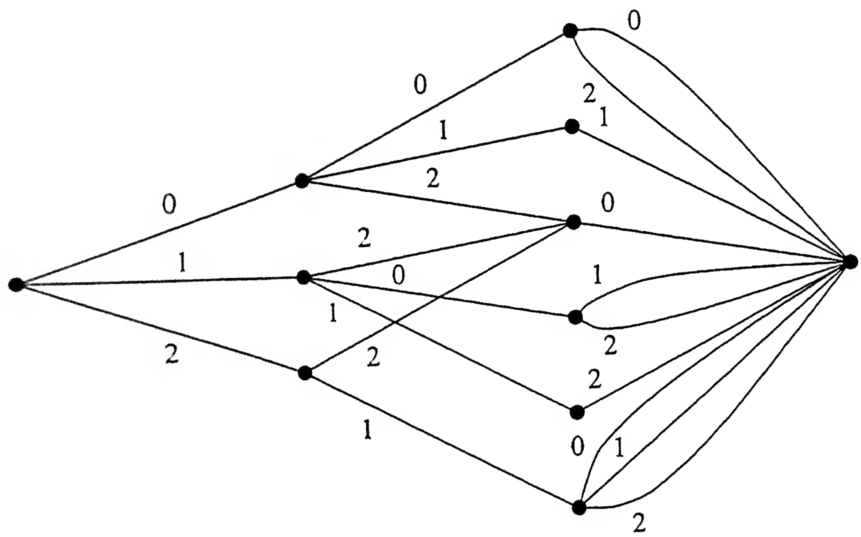

Figure 6.6.6: Minimal proper trellis for C_4 in Example 6.6.2

of vertices at each level of the minimal proper trellis are tabulated in Table 6.6.5. Consider the equivalent code C_5 obtained by permutation of the co-ordinate positions of C_0.
$C_5 = \{ 000, 110, 020, 120, 201, 011, 211, 121, 221\ 002\ 102, 202 \}$.
The minimal proper trellis for this code is shown in Figure 6.6.7. For this code the number of vertices at each level of the minimal proper trellis are tabulated in Table 6.6.6. From the tables it is seen that the minimal proper trellises for the codes C_1, C_2, C_4 and C_5 form the

Table 6.6.5: Number of vertices in the minimal proper trellis of C_4 in Example 6.6.2

Level	No. of vertices in the minimal proper trellis
0	1
1	3
2	6
3	1

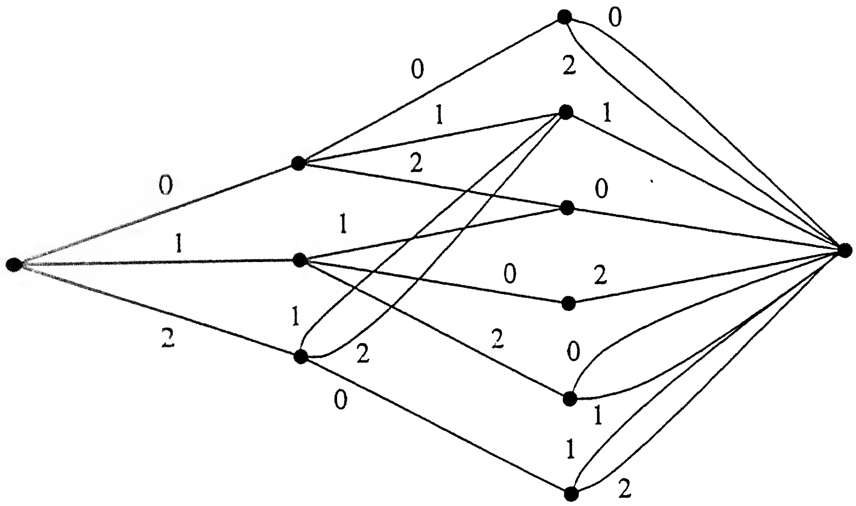

Figure 6.6.7: Minimal proper trellis for C_5 in Example 6.6.2

Table 6.6.6: Number of vertices in the minimal proper trellis of C_5 in Example 6.6.2

Level	No. of vertices in the minimal proper trellis
0	1
1	3
2	6
3	1

set of the optimum minimal proper trellis for the soft decoding of the code C_0. If the code C_2 is used for soft decoding, then if $r_0r_1r_2$ is the received word, the 3-tuple $r_1r_2r_0$ has to be considered, if the encoder for the block code is not redesigned for the code C_2.

6.7 Algorithm for Finding the Optimal Minimal Proper Trellis for a General Block Code

The scheme described in the previous section to obtain the minimal proper trellis for a general block code is summarized in the following algorithm.

(1) For the general block code, start constructing the code tree from the code table.

(2) At each level during the construction of the tree, check if two vertices at any level have subtrees isomorphic with weights.

(3) Merge the vertices at a level with isomorphic subtrees by replacing the vertices of the tree to be merged with a single vertex of the minimal proper trellis.

(4) All the edges, into the vertices of the tree to be merged will form the incoming edges into the vertex of the minimal proper trellis.

(5) The outgoing edges of the vertex of the minimal proper trellis will be the outgoing edges of the subtree (which is isomorphic with weights for all the vertices of the tree being merged) from level 0 to level 1.

(6) Continue steps (2) and (3) till the full code tree has been constructed.

(7) Merge all the vertices at the final level into a single vertex of the minimal proper trellis.

(8) Find all the equivalent codes for the given block code by permutations of the co-ordinate positions.

(9) Repeat steps (1) to (7), to obtain the minimal proper trellises for all the equivalent codes.

(10) Select the optimal minimal proper trellis from the set of the minimal proper trellises obtained.

(11) With the optimal minimal proper trellis permute the received word co-ordinates corresponding to the permutations for the equivalent code, or redesign the block encoder for the equivalent code.

6.8 Properties of the Minimal Proper Trellis

The following properties of the minimal proper trellis obtained, follow from the structure of the code tree. These properties also emphasize the equivalence between the reduced tree discussed in the previous chapter and the minimal proper trellis from the point of view of soft decoding.

Theorem 6.8.1 *The number of vertices of the minimal proper trellis obtained from the tree is equal to the number of partition sets* $V_{i,k}$ *of the tree at a level* i, $\forall\ i < n$.

Proof: From Definition 5.5.3, $V_{i,k}$ denotes the partition set at each level of the code tree. From Definition 6.6.3, Definition 6.6.4 and Theorem 6.6.1, the vertices of the minimal proper trellis are formed by merging the vertices of the tree inside the partition set.
Hence, $V_{i,k}$ is a vertex of the minimal proper trellis.
So, cardinality of the set $\{V_{i,k}\}$ $\forall\ i < n$, is the number of vertices of the minimal proper trellis at level i. □

Lemma 6.8.1 *The number of vertices of the optimal minimal proper trellis obtained from the tree is equal to the number of partition sets* $V_{i,k}$ *of the tree of the equivalent code at a level* i $\forall\ i < n$.

Proof: The result is a consequence of Theorem 6.8.1. □

Theorem 6.8.2 *At level* i, $0 < i < n$, *the number of edges into the vertex* $V_{i,k}$ *of the minimal proper trellis is equal to the cardinality of set* $V_{i,k}$.

Proof: From Definition 5.5.3, $V_{i,k}$ denotes the partition set at each level of the tree.
At each level i, consider $V_{i,k} = \{ v_{i,j_0}, v_{i,j_1}, \ldots, v_{i,j_{r-1}} \}$.
The set of edges of the tree into these vertices from level $i-1$ is, $\{ IV_{i,j_0}, IV_{i,j_1}, \ldots, IV_{i,j_{r-1}} \}$. From Theorem 6.6.1 and Definition 6.6.3, the vertex of the minimal proper trellis

$\mathrm{V}_{i,k}$ is formed by merging the vertices of the tree.
Hence, for the vertex of the minimal proper trellis $\mathrm{V}_{i,k}$, the edges into the vertex from previous level $i-1$ is, $\mathrm{IV}_{i,k,p} = \{ IV_{i,j_0}, IV_{i,j_1}, \ldots, IV_{i,j_{r-1}} \}$, $0 \leq p \leq r-1$.
Hence, cardinality of the set $\mathrm{IV}_{i,k,p}$ = cardinality of set $\mathrm{V}_{i,k}$. □

At level 0, there are no edges into the vertex of the trellis. At level n, the number of edges into the vertex of the trellis is equal to the number of vertices in the tree at level n, as all of these will be merged.

Lemma 6.8.2 *At level i, $0 < i < n$, the number of edges into the vertex $\mathrm{V}_{i,k}$ of the optimal minimal proper trellis is equal to the cardinality of set $\mathrm{V}_{i,k}$.*

Proof: The result is a consequence of Theorem 6.8.2. □

Theorem 6.8.3 *At any vertex of the minimal proper trellis $\mathrm{V}_{i,k}$, $i \neq n$ the number of edges out of the vertex is equal to the number of edges in the isomorphic subtree subtended at the corresponding vertex of the tree from level 0 to level 1.*

Proof: From Definition 5.5.3, $\mathrm{V}_{i,k}$ denotes the partition set at each level of the tree.
At a level i, $i \neq n$ consider $\mathrm{V}_{i,k} = \{ v_{i,j_0}, v_{i,j_1}, \ldots, v_{i,j_{r-1}} \}$.
From Definition 5.5.4, it follows that $\mathrm{T}_{v_{i,j_0}} \overset{\mathrm{W}}{\cong} \mathrm{T}_{v_{i,j_1}} \overset{\mathrm{W}}{\cong} \ldots \overset{\mathrm{W}}{\cong} \mathrm{T}_{v_{i,j_{r-1}}}$.
Hence, $\{ OV_{i,j_0,q} \} = \{ OV_{i,j_1,q} \} = \ldots = \{ OV_{i,j_{r-1},q} \}$.
From Theorem 6.6.1 and Definition 6.6.4, the vertex of the minimal proper trellis is formed by merging the vertices of the tree.
Hence, for the vertex of the minimal proper trellis $\mathrm{V}_{i,k}$,

$$\{ \mathrm{OV}_{i,k,q} \} = \{ OV_{i,j_0,q} \} = \{ OV_{i,j_1,q} \} = \ldots = \{ OV_{i,j_{r-1},q} \}.$$

But, cardinality of $\{ OV_{i,j_0,q} \}$ = the number of edges in the isomorphic subtree from level 0 to level 1. Hence, the result. □

Lemma 6.8.3 *At any vertex of the optimal minimal proper trellis $\mathrm{V}_{i,k}$, $i \neq n$ the number of edges out of the vertex is equal to the number of edges in the isomorphic subtree subtended at the corresponding vertex of the tree for the equivalent code, from level 0 to level 1.*

Proof: The result is a consequence of Theorem 6.8.3. □

Theorem 6.8.4 *Parallel transitions occur in the minimal proper trellis, when in the code tree. at level i, some vertices are in the same partition set* $\mathrm{V}_{i,k}$ *and at level* $(i-1)$*, these vertices have the same parent vertex.*

Proof: Parallel transitions occur in a trellis, when all the edges starting from a vertex at level $(i-1)$, again enter a same vertex at level i.
For this to occur, $\{\ \mathrm{OV}_{i-1,k_1,q}\ \} = \{\ \mathrm{IV}_{i,k_2,p}\ \}$.
That is, for the code tree vertex v_{i-1,k_1} and $\mathrm{V}_{i,k_2} = \{\ v_{i,j_0}, v_{i,j_1}, \ldots v_{i,j_{r-1}}\ \}$, $\{OV_{i-1,k_1,q}\} = IV_{i,j_0}, IV_{i,j_1}, \ldots IV_{i,j_{r-1}}\ \}$.
So for the tree the vertices in V_{i,k_2} have the same parent vertex at level $i-1$. □

Lemma 6.8.4 *Parallel transitions occur in the optimal minimal proper trellis, when in the code tree of the equivalent code. at level i, some vertices are in the same partition set* $\mathrm{V}_{i,k}$ *and at level* $(i-1)$*, these vertices have the same parent vertex.*

Proof: The result is a consequence of Theorem 6.8.4. □

6.9 Obtaining Optimal Minimal Improper Trellises for General Block Codes

The minimal trellis or the optimal minimal trellis need not be proper. Some scheme is required to obtain the improper trellis for the soft decoding of a general block code. This section considers the problem of obtaining the optimal minimal improper trellis for a general block code, for block length $n = 3$. For $n = 1$, the trellis for a block code is trivial and no permutations of the code positions are possible so no other equivalent code can exist. For n = 2 only one permutation is possible, but this does not affects the number of vertices. So these cases are not considered in the work.

The following characteristics of an improper trellis, used for the representation of block code words, can be stated.

(1) In an improper trellis more than one edge with the same label can emanate from a single vertex of the trellis.

(2) The improper trellis cannot be obtained from the tree, as in a tree no vertex can have two out going edges with the same label.

(3) In general, the minimal proper trellis for a nonlinear code requires more vertices at a given time index than the minimal improper trellis [51].

(4) In finding the minimal improper trellis the minimization of the vertex count at one time index may be incompatible with minimization of the vertex count at another time index [51].

Based on these problems, it is difficult to obtain the minimal improper trellis for a general code. In this thesis attention is restricted to the problem of finding the minimal improper trellis for general block codes with block length $n = 3$. The concept of code equivalence is used in finding the optimal minimal improper trellis.

Definition 6.9.1 *A general block code of length* 3, C_i, *is a* 3*-tuple* $c_0c_1c_2$.
Let, $\bar{S}_{i,0}$ *be the set of the symbols at co-ordinate position* 0 *in the code word, that is, corresponding to column* c_0 *in the code table.*
Let, $\bar{S}_{i,2}$ *be the set of the symbols at co-ordinate position* 2 *in the code word, that is, corresponding to column* c_2 *in the code table.*

Theorem 6.9.1 *A trellis in which the number of vertices at level* 1 *is equal to* $|\bar{S}_{i,0}|$ *and the number of vertices at level* 2 *is equal to* $|\bar{S}_{i,2}|$, *is the minimal improper trellis for the general block code* C_i *of block length* 3, *when the minimal trellis for the block code is improper.*

Proof: At level 0 and level 3, in the trellis only one vertex is present.
From level 0 to level 1 distinct edges are required for each symbol $\in \bar{S}_{i,0}$.
Hence, at level 1, the minimum number of vertices that can exist for the trellis of any code is $|\bar{S}_{i,0}|$.
Similarly, from level 2 to the final vertex at level 3, distinct edges are required for each symbol $\in \bar{S}_{i,2}$.
Hence, at level 2, the minimum number of vertices that can exist for the trellis of any code is $|\bar{S}_{i,2}|$.

Hence, the trellis obtained with this configuration of the vertices is the minimal trellis. Since, it is assumed that the minimal trellis is improper, the resultant trellis will be the minimal improper trellis. □

Based on this Theorem 6.9.1, the minimal improper trellis of a general block code of block length 3 can be found. If the minimal trellis for a block code is proper then, the scheme given in the earlier section can be used. If the nature of the minimal trellis is not known in advance then, the trellises obtained by both the approaches can be found and from these, the minimal trellis can be selected.

Based on Theorem 6.9.1, the Theorem 1 of Muder [61], can be modified as follows.

Theorem 6.9.2 *Every general block code of block length* 3, *has a minimal trellis and the two minimal proper trellises or the two minimal improper trellises, which ever exist, for the same code are isomorphic.*

Proof: This result follows from Theorem 6.9.1. □

Theorem 6.9.3 *For a general block code of block length* 3, *the minimal improper trellis, corresponding to the equivalent code C_i, which results in the minimization of*

$$|\bar{\mathrm{S}}_{i,0}| \text{ and } |\bar{\mathrm{S}}_{i,2}|,$$

is the optimal minimal improper trellis for the code.

Proof: From Theorem 5.5.3 and the Lemma to this theorem, the equivalent code can be used for the soft decoding of a code and can results in different minimal improper trellises for a code.

From Definition 6.4.6, minimization of the vertices of the trellises at level 1 and level 2 is equivalent to minimization of

$$|\bar{\mathrm{S}}_{i,0}| \text{ and } |\bar{\mathrm{S}}_{i,2}|.$$

Hence, the trellis for the equivalent code is the optimal minimal improper trellis. □

Theorem 6.9.4 *For a general block code of block length* 3, *if all the symbols from the expanded channel signal constellation S' are used at all the three co-ordinate locations, then the minimal improper trellis is the optimal minimal improper trellis for the code.*

Proof: If for a code all the symbols from the expanded channel signal constellation S' are used at all the three co-ordinate locations, then

$$\forall i |\bar{\mathrm{S}}_{i,0}| = |\bar{\mathrm{S}}_{i,2}| = n'.$$

Hence, all the equivalence codes result in the same trellis.
So the minimal improper trellis will be the optimal minimal improper trellis for the code. □

Example 6.9.1 Consider the Example code for a BCM scheme, $C_0 = \{000, 011, 020, 021, 102, 110, 112, 121, 122\ 200\ 201, 202\}$, for which the minimal proper trellis was obtained in the previous section.

For this code, $\bar{\mathrm{S}}_{0,0} = \{0, 1, 2\}$ and $\bar{\mathrm{S}}_{0,2} = \{0, 1, 2\}$.
Hence, $|\bar{\mathrm{S}}_{0,0}| = 3$ and $|\bar{\mathrm{S}}_{0,2}| = 3$.
The minimal trellis for C_0 can be obtained based on Theorem 6.9.1 and it is improper.
Also, from Theorem 6.9.4, for this code the optimal minimal improper trellis is also the minimal improper trellis for C_0. The optimal minimal improper trellis is shown in Figure 6.9.1.
The optimal minimal improper trellis has 3 vertices at level 1 and level 2, and it is also the optimal minimal trellis for the code C_0.

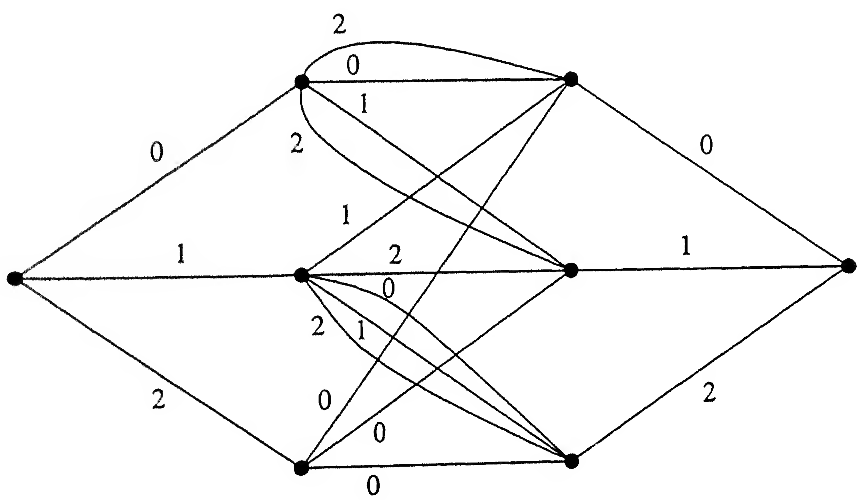

Figure 6.9.1: The optimal minimal improper trellis for C_0 in Example 6.9.1

Example 6.9.2 Consider the following block code [51, Appendix] $C_0 = \{ 115, 122, 123, 213, 214, 215, 222, 223, 224, 313, 314, 316, 321, 324, 326, 414, 416, 421, 426 \}$.

The equivalent codes for C_0 are obtained as follows.

$C_1 = \{ 151, 122, 132, 231, 241, 251, 222, 232, 242, 331, 341, 361, 312, 342, 362, 441, 461, 412, 462 \}$.

$C_2 = \{ 151, 221, 231, 132, 142, 152, 222, 232, 242, 133, 143, 163, 213, 243, 263, 144, 164, 214, 264 \}$.

$C_3 = \{ 115, 212, 213, 123, 124, 125, 222, 223, 224, 133, 134, 136, 231, 234, 236, 144, 146, 241, 246 \}$.

$C_4 = \{ 511, 212, 312, 321, 421, 521, 222, 322, 422, 331, 431, 631, 132, 432, 632, 441, 641, 142, 642 \}$.

$C_5 = \{ 511, 221, 321, 312, 412, 512, 222, 322, 422, 313, 413, 613, 123, 423, 623, 414, 614, 124, 624 \}$.

The $\bar{S}_{i,0}$, $|\bar{S}_{i,0}|$, $\bar{S}_{i,2}$ and $|\bar{S}_{i,2}|$ for the equivalent codes are given in Table 6.9.1.

The minimal improper trellis corresponding to the equivalent codes C_1 or C_2, shown in

Table 6.9.1: Table for equivalent codes in Example 6.9.2

C_i	$\bar{S}_{i,0}$	$\lvert\bar{S}_{i,0}\rvert$	$\bar{S}_{i,2}$	$\lvert\bar{S}_{i,2}\rvert$
C_0	$\{1,2,3,4\}$	4	$\{1,2,3,4,5,6\}$	6
C_1	$\{1,2,3,4\}$	4	$\{1,2\}$	2
C_2	$\{1,2\}$	2	$\{1,2,3,4\}$	4
C_3	$\{1,2\}$	2	$\{1,2,3,4,5,6\}$	6
C_4	$\{1,2,3,4,5,6\}$	6	$\{1,2\}$	2
C_5	$\{1,2,3,4,5,6\}$	6	$\{1,2,3,4\}$	4

Figure 6.9.2 and Figure 6.9.3 will be the optimal minimal improper trellis for this code. Both the improper trellises are isomorphic.

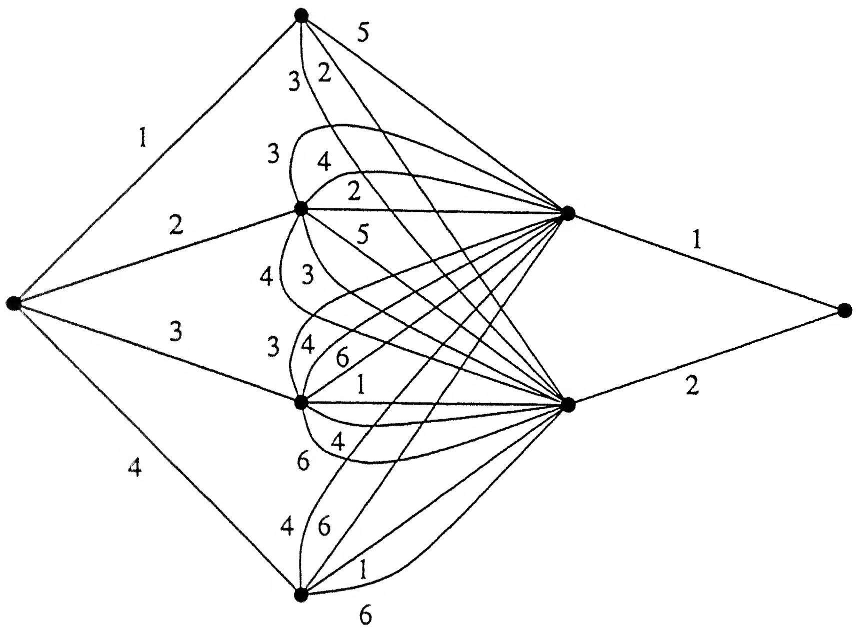

Figure 6.9.2: The minimal improper trellis for C_1 in Example 6.9.2

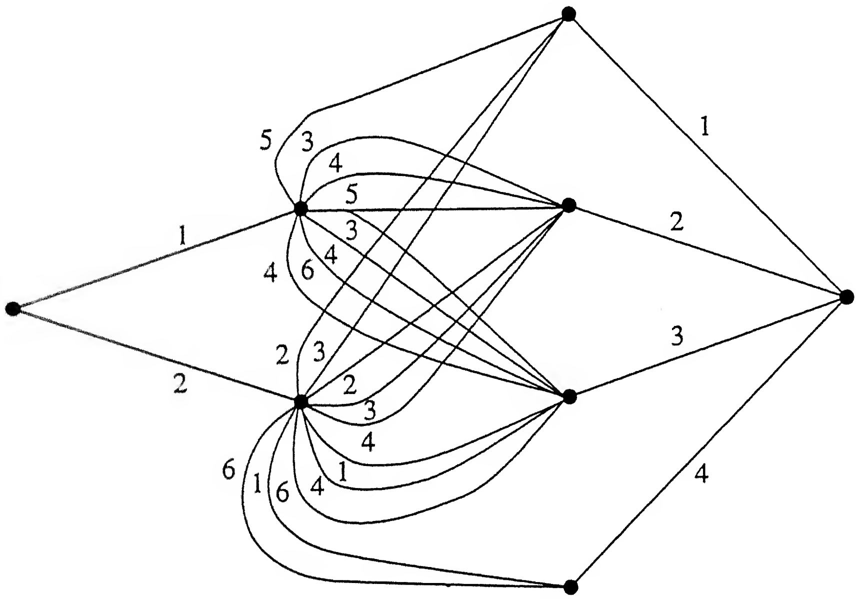

Figure 6.9.3: The minimal improper trellis for C_2 in Example 6.9.2

6.10 Algorithm for Finding the Optimal Minimal Improper Trellis for a General Block Code for n = 3

(1) Obtain the minimal improper trellis for the general block code of length $n = 3$, by having one initial vertex one final vertex, $|\bar{S}_{i,0}|$ vertices at level 1 and $|\bar{S}_{i,2}|$ vertices at level 2, and, connecting and labeling the edges based on the code words.

(2) For the block code, find out if all the symbols of the expanded channel signal constellation are used at all the co-ordinate positions of the code words. If yes, then the minimal improper trellis is the optimal minimal improper trellis for the code.

(3) Find all the equivalent codes (totally 5 more than the original block code).

(4) Obtain $|\bar{S}_{i,0}|$ and $|\bar{S}_{i,2}|$ for all the equivalent codes.

(5) Select the equivalent code which minimizes $|\bar{S}_{i,0}|$ and $|\bar{S}_{i,2}|$. The minimal improper trellis for this code obtained using step (1), is the optimal minimal improper trellis for the code.

(6) With the optimal minimal improper trellis permute the received word co-ordinates corresponding to the permutations for the equivalent code.

6.11 Examples

In this section the optimal minimal proper and improper trellises are obtained for the general codes found in Chapters 3 and 4.

Example 6.11.1 The optimal minimal trellis for the code in Example 3.3.1 is shown in Figure 6.11.1.

Example 6.11.2 The optimal minimal proper trellis for the code { 0000, 1120, 1102, 1311, 3111, 0231, 2031, 0213, 2013, 2200, 0022, 3320, 3302, 1333, 3133, 2222 }, equivalent to the code in Example 3.10.1 is shown in Figure 6.11.2.

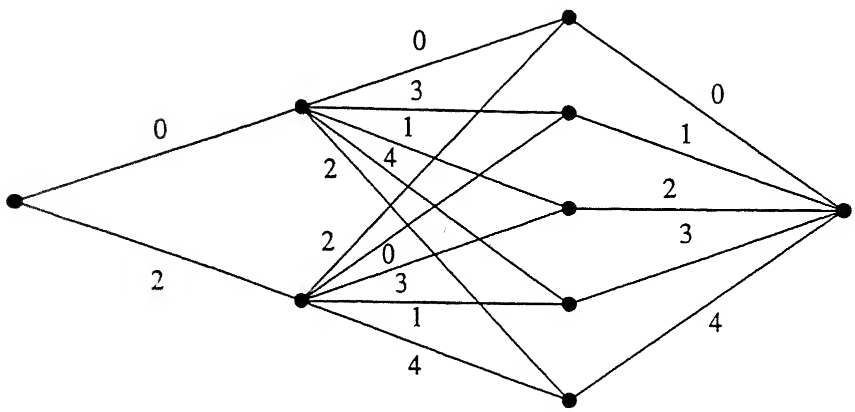

Figure 6.11.1: The optimal minimal trellis for the code in Example 6.11.1

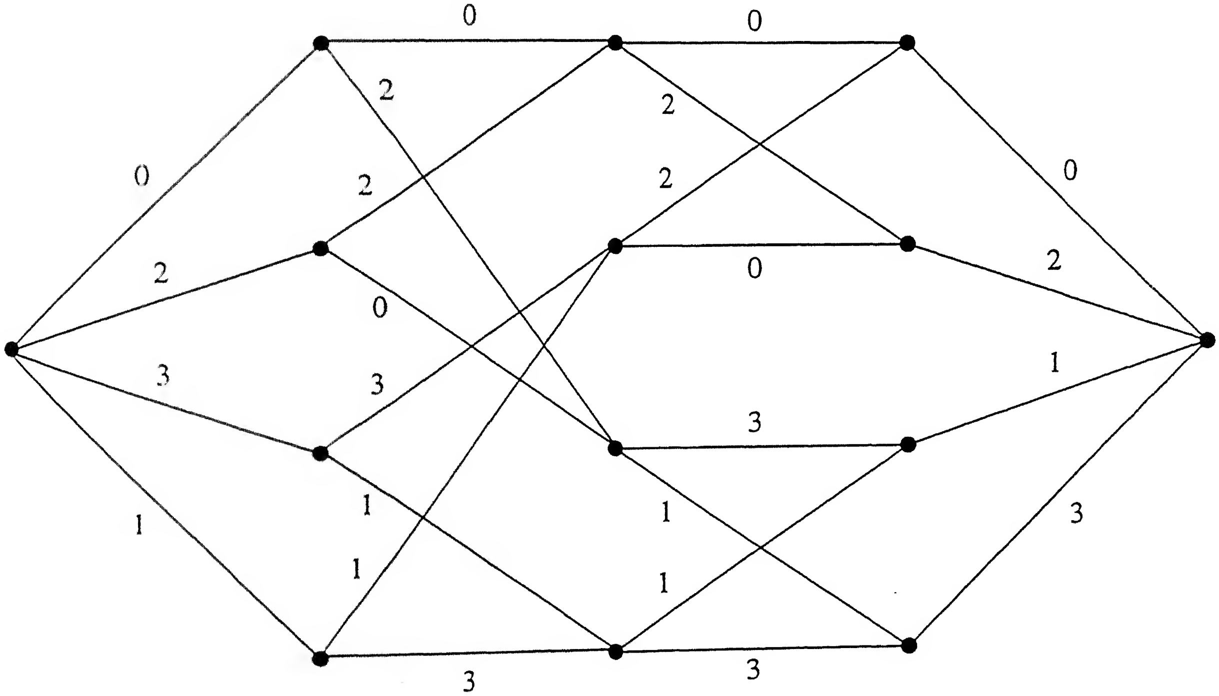

Figure 6.11.2: The optimal minimal proper trellis for the code in Example 6.11.2

Example 6.11.3 The optimal minimal trellis for the code in Example 3.10.2 is shown in Figure 6.11.3.

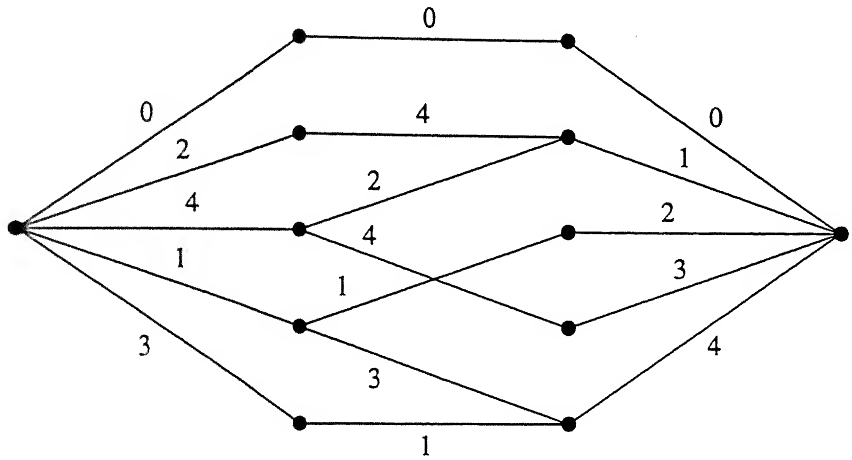

Figure 6.11.3: The optimal minimal trellis for the code in Example 6.11.3

Example 6.11.4 The optimal minimal proper trellis for the code in Example 4.8.1 is shown in Figure 6.11.4.

Example 6.11.5 The optimal minimal proper trellis for the code in Example 4.8.2 is shown in Figure 6.11.5.

6.12 Concluding Remarks

This chapter proposes the use of the optimal minimal trellis for the soft decoding of general block codes. A scheme of using equivalent codes, for obtaining the optimal minimal proper trellis for the soft decoding of general block codes, used with BCM is discussed. A condition for obtaining the optimal minimal improper trellis, and hence the optimal minimal trellis, for general block codes of block length 3, is obtained. The optimal minimal proper trellis for a general code is obtained from the reduced code tree of an equivalent code. Properties of the minimal proper trellises as implied from the nature of the code tree are also discussed. It is shown that every general block code of block length 3, has a minimal trellis and the two minimal proper trellises or the two minimal improper trellises, which ever exist, for the same code are isomorphic.

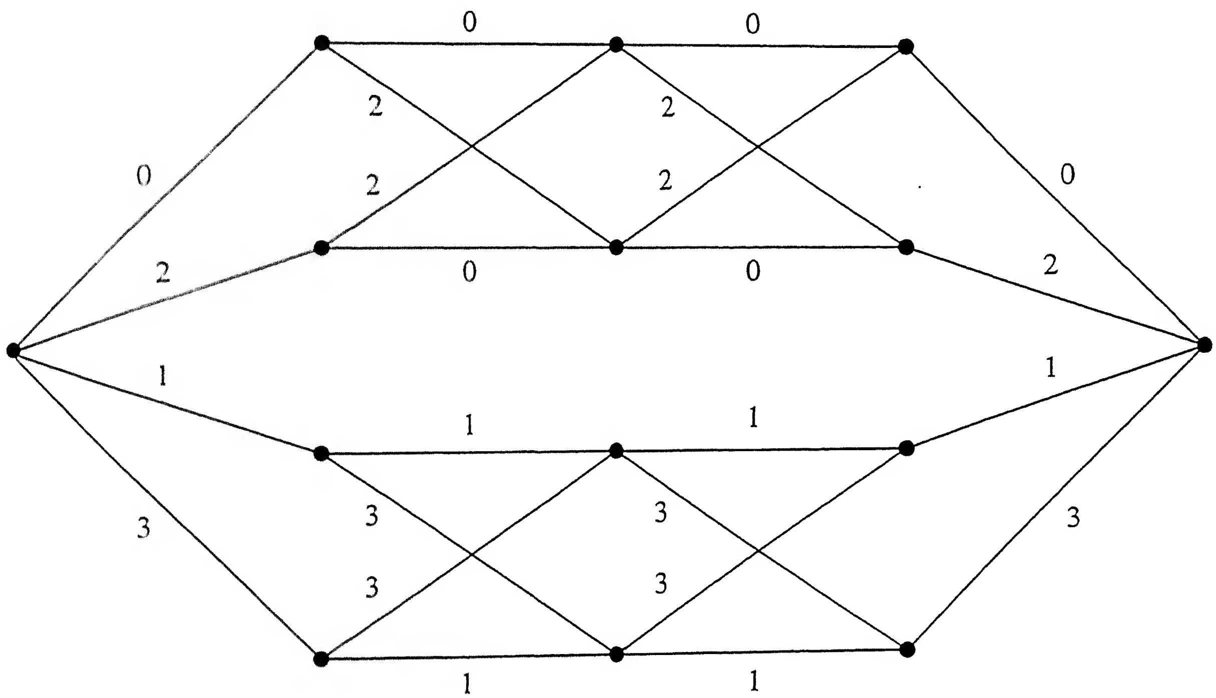

Figure 6.11.4: The optimal minimal trellis for the code in Example 6.11.4

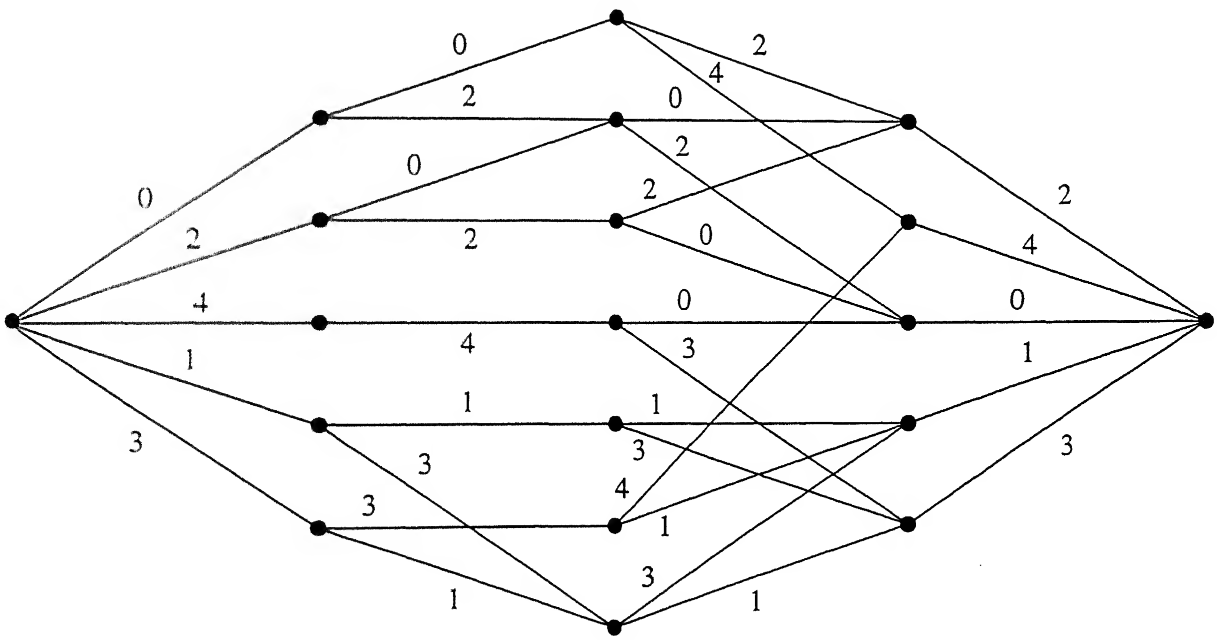

Figure 6.11.5: The optimal minimal trellis for the code in Example 6.11.5

Chapter 7

Concatenated BCM for Obtaining General Block Codes

7.1 Introduction

This chapter uses general block codes of short block lengths, to obtain block codes of larger block lengths, using concatenation. Both the inner and the outer codes correspond to a BCM scheme. The inner block code is considered as a virtual expanded channel signal constellation by the outer block code. The scheme considers all block codes of a fixed block length. The code search, encoding and soft decoding complexity, of these codes is reduced as compared to directly using a block code of long length.

Since, a virtual channel signal constellation is arbitrary, the structured distance approach discussed in Chapters 3 and 4, has to be used for obtaining codes and for the implementation of the encoder. Soft decoding, using a reduced tree or an optimal minimal trellis discussed in Chapters 5 and 6 can be employed with the concatenation scheme.

The chapter begins with a brief overview of some recent relevant references. Some considerations regarding the optimum signal constellation for coded modulation schemes are discussed. The scheme for the encoding of concatenated BCM-BCM codes is presented. The advantages of the scheme, in the implementation of the soft decoder, are discussed. The results are summarized in Appendix B. Finally, the chapter ends with some concluding remarks.

7.2 Background and Preliminaries

Concatenated codes [58] were used for obtaining binary codes, by combining other binary codes. Concatenation is also used with coded modulation, and various schemes have been proposed in the literature, which use a combination of schemes.

A brief outline of some recent literature using concatenated codes is discussed here. Rajpal et al. [65, 66] have used binary convolutional codes with good free branch distances as outer codes and block MPSK modulation codes as inner codes. Pellizzoni and Spalvieri [62] discuss the encoding and decoding for binary multilevel coset codes obtained by concatenating convolutional outer codes and Reed-Muller inner codes. Deng and Costello [22] give a concatenated scheme with TCM as inner code and Reed-Solomon codes as outer codes. Herzberg et al. [39] discuss a concatenated scheme with lattice codes as inner codes and Reed-Solomon codes with hard-decision decoding as outer codes.

Generally, the motivation behind concatenation is to obtain a resultant code which is different from the component codes and has various desirable properties. In this chapter concatenation is used to obtain block codes of long length using block codes of shorter lengths, and to also simplify the code search, encoding and the decoding complexity of general block codes. Concatenation provides a trade-off between time (length of the block code n) and space (the number of signals in the virtual channel signal constellation).

7.3 Motivating Factors

(1) It is desirable to obtain the optimum signal constellation, which is efficient for use with coded modulation. It is also necessary to consider the practical implementability of such a signal constellation using a modulation scheme.

(2) In a BCM scheme, it is advisable to use codes of short block lengths, since in general, the complexity of the code search increases with the block length n.

(3) From Section 5.6.2, for parallel implementation of a soft decoder using a reduced tree, it is better to use codes of short block length, since parallelism is achieved at the cost of increase in data memory requirement.

(4) From Section 6.9. it follows that an optimal minimal trellis can be found for all general block codes of block length 3. So a decoding scheme using the Viterbi algorithm will be efficient, using codes of block length 3.

(5) Some scheme is required to obtain codes for BCM schemes of large block lengths, in a systematic manner using codes of short block lengths.

7.4 The Optimum Signal Constellation and Virtual Signal Constellations

In Chapter 2, an arbitrary channel signal constellation S' has been defined by the matrix $\mathbf{d_S}$. The matrix $\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\ldots\times\mathbf{S}(\mathbf{n-times})}$, gives the Euclidean distances between all the n-tuples of the signals from S'. It represents the discrete Euclidean space.

It is interesting to consider the structure of the matrix $\mathbf{d_S}$, which will result in an good code, that is, which is efficient for coded modulation. Here, the problem of coding has to be posed in a different manner, as now, it is of interest to optimize the signal constellation. so that good codes, that is. codes which maximize $d_{\min}$ can be found. It is also of interest to find the corresponding modulation scheme which will provide the optimum signal constellation.

Theorem 7.4.1 *The matrix* $\mathbf{d_S}$ *is symmetric.*

Proof: The elements $d_{i,j}$ of $\mathbf{d_S}$ are such that $d_{i,j} = d(s_i, s_j)$, where, s_i, $s_j \in S'$ and $d(\ )$, denotes the Euclidean distance.
From Theorem 2.2.1, the set S' with distance, d forms a metric space.
Hence, $d_{i,j} = d_{j,i}$,
$d_{i,j} > 0$, if $i \neq j$ and $d_{i,j} = 0$ if $i = j$.
Hence, the matrix $\mathbf{d_S}$ is symmetric. □

Lemma 7.4.1 *The elements* $d_{i,j}$ *of the matrix* $\mathbf{d_S}$ *are such that,*

$$d_{i,j} \leq d_{i,k} + d_{k,j}.$$

Proof: This result follows from Theorem 2.2.1. □

Theorem 7.4.2 *$\forall\ i,\ j.$ such that, $d_{i,j} \in \mathbf{d_S}$, $d_{i,j} \leq K$. (For the signal constellations $K = 2$ is assumed)*[1].

Proof: S' is an expanded channel signal constellation. For BCM the channel is assumed to be power-limited. For a n'-dimensional channel signal constellation all the signals due to the power constraint will lie inside a n'-dimensional sphere of radius K.
This implies that $d_{i,j} \leq K$. □

Theorems 7.4.1, 7.4.2 and Lemma 7.4.1 characterize the matrix $\mathbf{d_S}$. To achieve the objective of maximization of the minimum d_{min}, it is necessary to consider the non-zero distances in $\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\ldots\times\mathbf{S}(\mathbf{n-times})}$. This has to result from a proper selection of $\mathbf{d_S}$.

Theorem 7.4.3 *The non-zero Euclidean distances in $\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\ldots\times\mathbf{S}(\mathbf{n-times})}$ are maximized iff $d_{i,j} = 2\ \forall\ i \neq j$ and $d_{i,j} \in \mathbf{d_S}$.*

Proof: From Theorem 2.3.2,

$$\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\ldots\times\mathbf{S}(\text{n-times})} = \mathbf{d_S}(Ed)\mathbf{d_S}\ldots(Ed)\mathbf{d}_{\mathbf{S}(\text{n-times})}.$$

From Definitions 2.3.1 and 2.3.2, it follows that,
for an element, $d_{i,k}\ (Ed)\ d_{j,l} = \sqrt{d_{i,k}^2 + d_{j,l}^2} = d_{ij,kl}$.
Hence, maximization of the element $d_{ij,kl}$, implies the maximization of $d_{i,k}$ and $d_{j,l}$.
But $d_{i,k}$, $d_{j,l} \in \mathbf{d_S}$, such that $d_{i,k} \neq 0$ and $d_{j,l} \neq 0$.
From Theorem 7.4.2,
Maximum $d_{i,k} = 2 \quad \forall\ i \neq k$ and $d_{i,k} \in \mathbf{d_S}$, and
Maximum $d_{j,l} = 2 \quad \forall\ j \neq l$ and $d_{j,l} \in \mathbf{d_S}$.
Also, if any of these (say) $d_{i,k} \neq 2$, then this implies that $d_{ij,kl} <$ Maximum $d_{ij,kl}$.
Hence, the non-zero Euclidean distances in $\mathbf{d}_{\mathbf{S}\times\mathbf{S}\times\ldots\times\mathbf{S}(\mathbf{n-times})}$ are maximized iff $d_{i,j} = 2\ \forall\ i \neq j$ and $d_{i,j} \in \mathbf{d_S}$. □

The result obtained in Theorem 7.4.3 conforms with the nature of matrix $\mathbf{d_S}$ as given in Theorem 7.4.1, 7.4.2 and Lemma 7.4.1. But in practice the signal constellation might not exist.

Example 7.4.1 Consider the 2-PSK signal constellation shown in Appendix A.

[1]Signals are assumed to be points inside the unit n'-dimensional sphere.

The matrix $\mathbf{d_S}$ for this signal constellation is as follows,

$$\mathbf{d_S} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}. \tag{7.4.1}$$

Consider sequences of length 2.

$$\mathbf{d_{S\times S}} = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 4 & 2 \\ 2 & 4 & 0 & 2 \\ 4 & 2 & 2 & 0 \end{bmatrix}. \tag{7.4.2}$$

For sequences of length 3,

$$\mathbf{d_{S\times S\times S}} = \begin{bmatrix} 0 & 2 & 2 & 4 & 2 & 4 & 4 & 8 \\ 2 & 0 & 4 & 2 & 4 & 2 & 8 & 4 \\ 2 & 4 & 0 & 2 & 4 & 8 & 2 & 4 \\ 4 & 2 & 2 & 0 & 8 & 4 & 4 & 2 \\ 2 & 4 & 4 & 8 & 0 & 2 & 2 & 4 \\ 4 & 2 & 8 & 4 & 2 & 0 & 4 & 2 \\ 4 & 8 & 2 & 4 & 2 & 4 & 0 & 2 \\ 8 & 4 & 4 & 2 & 4 & 2 & 2 & 0 \end{bmatrix}. \tag{7.4.3}$$

These matrices have maximum number of elements. which are maximum. No other matrix better than these exists. Hence, $\mathbf{d_S}$ is the optimum signal constellation for coded modulation. when two signals are required in a single dimension. This signal constellation is physically realizable.

If a signal constellation in higher dimension is required, for example, say 2-dimensions with 4 signals. then the matrix for this to satisfy Theorem 7.4.3 has to be,

$$\mathbf{d_S} = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 2 & 2 \\ 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{bmatrix}. \tag{7.4.4}$$

Such a signal constellation cannot practically exist. Hence the optimum signal constellation. in this case, is non-realizable by using the existing modulation techniques.

In practice a **virtual channel signal constellation** can be used to obtain the optimum signal constellation. A virtual channel signal constellation can be considered to be code

words obtained using another coded modulation scheme, which can be TCM or BCM. In this thesis, the virtual channel signal constellation corresponds to codes of an BCM scheme obtained from an actual signal constellation. The signals of a virtual channel signal constellation are sequences of signals from another actual channel signal constellation. Hence, in general, a virtual channel signal constellation is an arbitrary channel signal constellation. The channel carries the signals from the actual signal constellation. The virtual channel signal constellation is only used intermediately for increasing the efficiency of code search, encoding and decoding of the coded modulation scheme. The objective of coding for the block code of the virtual channel signal constellation is to obtain a signal constellation with some specific distribution of distances. It is the block code which uses the virtual channel signal constellation, that finally provides the $d_{\min}$ required for the concatenated scheme, using the efficient virtual channel signal constellation.

The implementation of a BCM scheme using virtual channel signal constellation which is in turn obtained from a BCM scheme, corresponds to concatenated BCM-BCM scheme. Codes can be obtained and encoding can be performed with the virtual channel signal constellations using the general frame work developed in Chapters 3 and 4. This chapter considers encoding and decoding schemes using the virtual channel signal constellation treated as a BCM scheme.

Example 7.4.2 Consider the 2-PSK signal constellation shown in Appendix A.

Consider a block code using this channel signal constellation having 4 code words. If these

Table 7.4.1: The virtual channel signal constellation discussed in Example 7.4.2

Signal of the virtual signal constellation	Code word of the BCM scheme
0	00
1	02
2	20
3	22

four code words are considered to be signals of a virtual channel signal constellation. Then,

$$\mathbf{d_S} = \begin{bmatrix} 0 & 2 & 2 & \sqrt{8} \\ 2 & 0 & \sqrt{8} & 2 \\ 2 & \sqrt{8} & 0 & 2 \\ \sqrt{8} & 2 & 2 & 0 \end{bmatrix}. \tag{7.4.5}$$

An actual signal constellation with such a $\mathbf{d_S}$ can not exist. But an virtual channel signal constellation, by the selection of an appropriate BCM scheme can result in such a signal constellation. This chapter considers BCM schemes using virtual channel signal constellations. The encoding and decoding for such concatenated BCM-BCM schemes are discussed in the succeeding sections.

7.5 Systematic Concatenation of BCM Codes to Obtain a General Block Code of Long Block Length

On the basis of the discussions carried out, it is desirable to incorporate the following features in the concatenated schemes.

(1) The virtual channel signal constellation is a block code obtained from a BCM scheme, hence, concatenated BCM-BCM schemes are considered.

(2) The concatenated scheme can be obtained by the concatenation of two or more BCM schemes. When more than two codes are concatenated for the o_i^{th} scheme, the concatenated scheme till the o_{i-1}^{th} scheme, will be considered to be a virtual channel signal constellation.

(3) The concatenated scheme is systematic, in the sense that each component codes of the BCM schemes used are of the same block length n.

(4) Based on decoding considerations the block length n is assumed to be equal to 3.

(5) The concatenated scheme can have redundancy in space only, or in time and space.

Since, block codes of length 3 are considered for concatenation, if the concatenated scheme uses two BCM schemes o_0 and o_1, then the resulting block code will be of block length 9. If a three stage concatenation is considered with BCM schemes o_0, o_1 and o_2, then the block code of length 27 is obtained. In general for a q-stage concatenated scheme, $q > 2$, using schemes o_0, o_1, ... o_{q-1}, the block code of length 3^q, will be obtained. For a scheme the number of code words and hence the number of data words, depend on the selection of the individual BCM schemes, whether redundancy is added in space only or in space and time.

The code search is simplified, since instead of searching for a block code of some length n, where $n = 3^q$ and $q > 2$, it is necessary to search for codes of length 3. At each stage a block code can be obtained using the schemes described in Chapters 3 and 4. The block code for stage o_0, the BCM scheme uses the actual signal constellation selected for the application. The block code at any stage i, $0 < i \leq (q-1)$, uses the virtual channel signal constellation provided by the stage $(i-1)$. The block code at the o_{q-1}^{th} stage, has to be selected to give sufficient number of code words and the required d_{min} for the concatenated BCM scheme. The block codes at the stages o_0, o_1, ..., o_{q-2} are selected to provide a virtual channel signal constellation with proper number of signals and a proper Euclidean distance distribution, for the succeeding stage. In this manner the complexity of the code search is reduced in a concatenated scheme.

The block code for a concatenated BCM-BCM scheme is represented as $(B', S', n, |C|, d_{\text{min}})$. Where,

B' - is the base signal constellation,
S' - is the actual expanded channel signal constellation,
n - is the block length ($= 3^q$),
$|C|$ - is the number of code words and
d_{min} - is the minimum Euclidean distance between the code words.

For each stage in the concatenated scheme the block codes of length 3 are represented as $(|V|, d_{\text{min}})_0$ — $(|V|, d_{\text{min}})_1$ — ... — $(|V|, d_{\text{min}})_{q-2}$ — $(|C|, d_{\text{min}})_{q-1}$.
Where,
$|V|$ - is the number of code words at a stage, that is the number of signals in the virtual channel signal constellation provided for the succeeding stage and
d_{min} - is the minimum Euclidean distance between the code words at a stage, that is the

minimum Euclidean distance provided by the virtual channel signal constellation for the succeeding stage.

Concatenation simplifies the implementation of the block encoder, as encoding can proceed in stages in a concatenated BCM-BCM scheme. Each stage has to deal with 3-tuples of signals from the signal constellation provided by the previous stage. The block diagram for the block encoder of a concatenated BCM-BCM scheme is shown in Figure 7.5.1. Starting

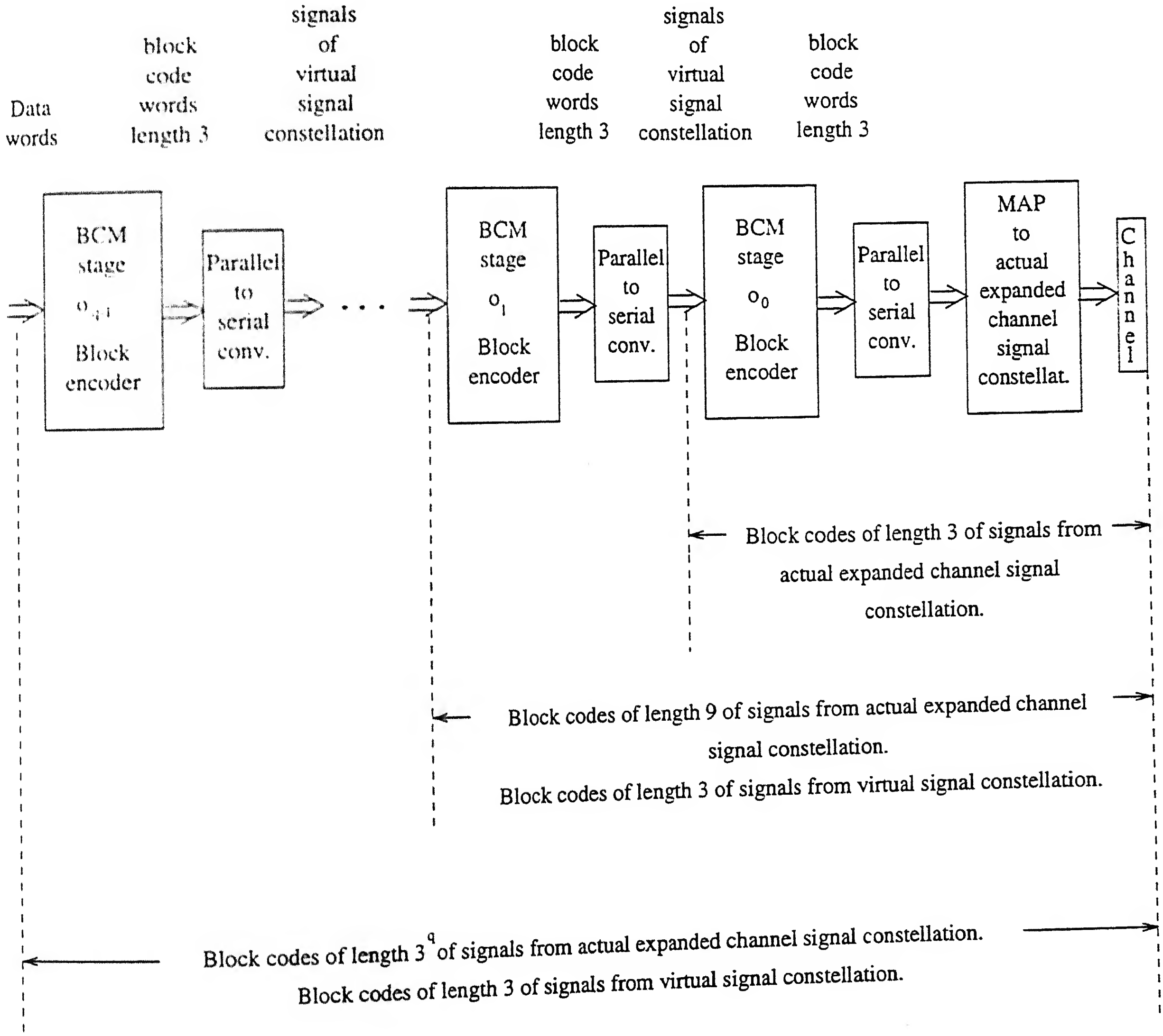

Figure 7.5.1: Block encoder for concatenated BCM-BCM schemes

from stage o_{q-1}, the encoder, which is basically a map from the data word to the code word, outputs a 3-tuple of symbols from the virtual channel signal constellation. Since, each virtual channel signal constellation is also a block code, each symbol is again block encoded by the succeeding stage, this continues till the first stage where finally the symbols are mapped to the signals of the actual expanded channel signal constellation.

Example 7.5.1 Consider a concatenated BCM-BCM scheme using the the 2-PSK signal constellation as the actual channel signal constellation.

A three stage concatenation of BCM schemes is employed.
$(4, 8.0)_0 - (16, 16.0)_1 - (256, 32.0)_2$.
The concatenated code can be represented by (—, 2-PSK, 27, 256, 32.0) code.
For this code redundancy is present in space and time. The block diagram for the block encoder is shown in Figure 7.5.2.

Concatenated BCM-BCM schemes can be classified into the following three types depending on the nature of the block codes.

Type 1 codes: The BCM scheme resulting in the virtual channel signal constellation is such that the cardinality of the set of Euclidean distances between signals of the virtual channel signal constellation is small. The BCM scheme using the virtual channel signal constellation, uses Euclidean distances and soft decoding.

Type 2 codes: The BCM scheme resulting in the virtual channel signal constellation is such that the cardinality of the set of Euclidean distances between signals of the virtual channel signal constellation is small. The BCM scheme using the virtual channel signal constellation, uses Hamming distances and hard decoding.

Type 3 codes: The BCM scheme resulting in the virtual channel signal constellation is such that the cardinality of the set of Euclidean distances between signals of the virtual channel signal constellation is not small. The BCM scheme using the virtual channel signal constellation, uses Euclidean distances and soft decoding.

The examples considered here, are of type 3 codes, which are more general than the other types of codes.

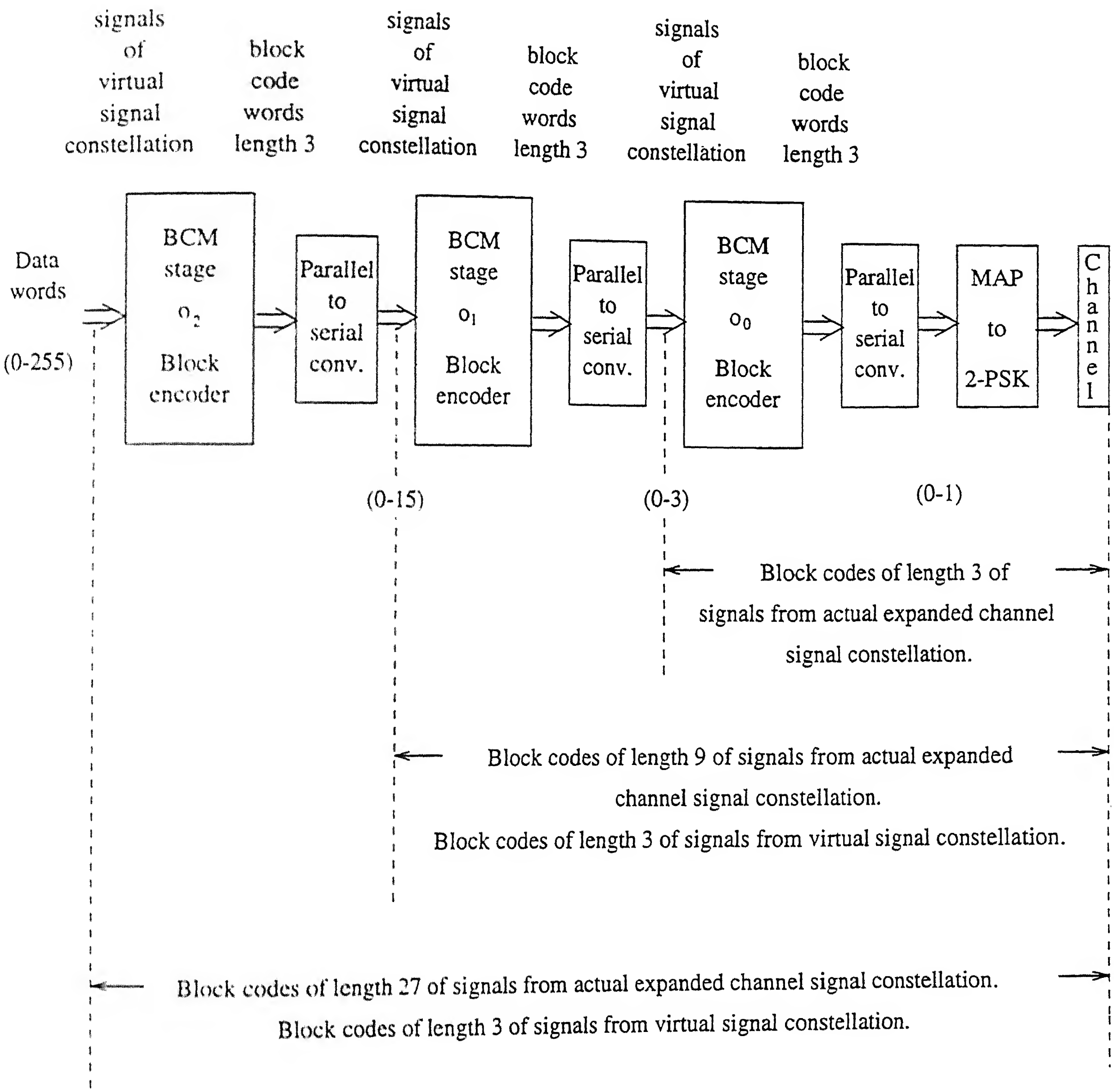

Figure 7.5.2: Block encoder for concatenated BCM-BCM scheme of Example 7.5.1

7.6 Soft Decoding of Concatenated BCM Schemes

Concatenated BCM schemes uses block codes of short length 3. Soft decoding of concatenated BCM schemes can be performed in stages. A reduced tree based soft decoder discussed

in Chapter 5 can be used, or the optimal minimal trellis can be obtained as described in Chapter 6 and the Viterbi algorithm can be used. Independent of the scheme used, the general block diagram of a soft decoder is shown in Figure 7.6.1. As soon as the soft decoder

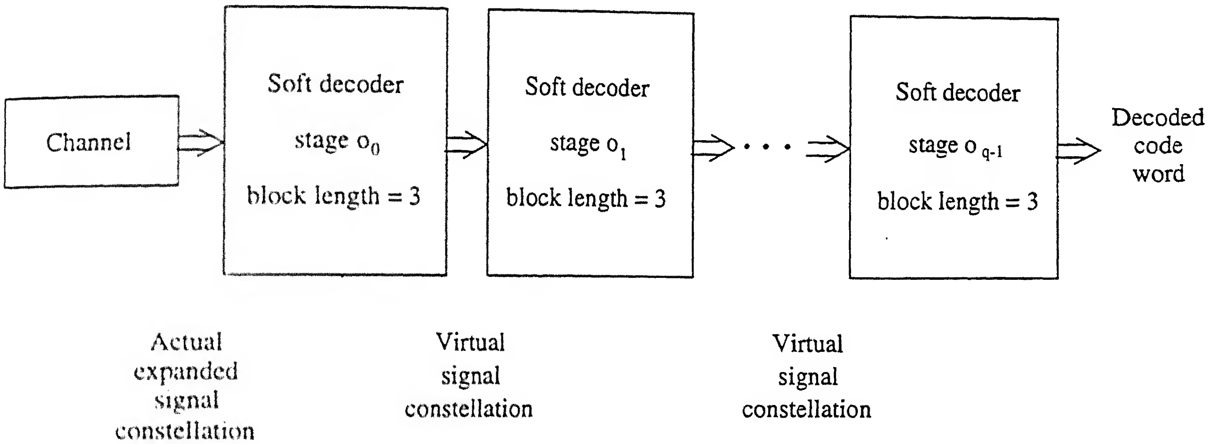

Figure 7.6.1: Soft decoder for concatenated BCM-BCM schemes

at stage o_0 outputs a decoded code word of length 3, the soft decoding of stage o_1 can start, this proceeds till from stage o_{q-1} the finally decoded code word is obtained. Simultaneously each stage can proceed with the soft decoding of the subsequent received words. In this way the decoding complexity is reduced. Decoding at various stages can be done in parallel and the soft decoder at each stage is also optimized by having block codes of length 3.

If a reduced tree based staged soft decoder is used, then parallelism can be utilized for having fast soft decoders. At various stages processing elements can be assigned and soft decoding can proceed in parallel even inside a stage.

If an optimal minimal trellis is used then for each stage depending on the code the trellis will be used by the Viterbi algorithm to perform soft decoding.

Example 7.6.1 Consider the concatenated BCM-BCM scheme using the the 2-PSK signal constellation as the actual channel signal constellation, discussed in Example 7.5.1.

A three stage soft decoder for the concatenated BCM scheme is shown in Figure 7.6.2.

The soft decoder at stage o_0, will have to perform soft decoding 9 times of the received symbols of length 3 and the maximum number of states at this stage can be 4.

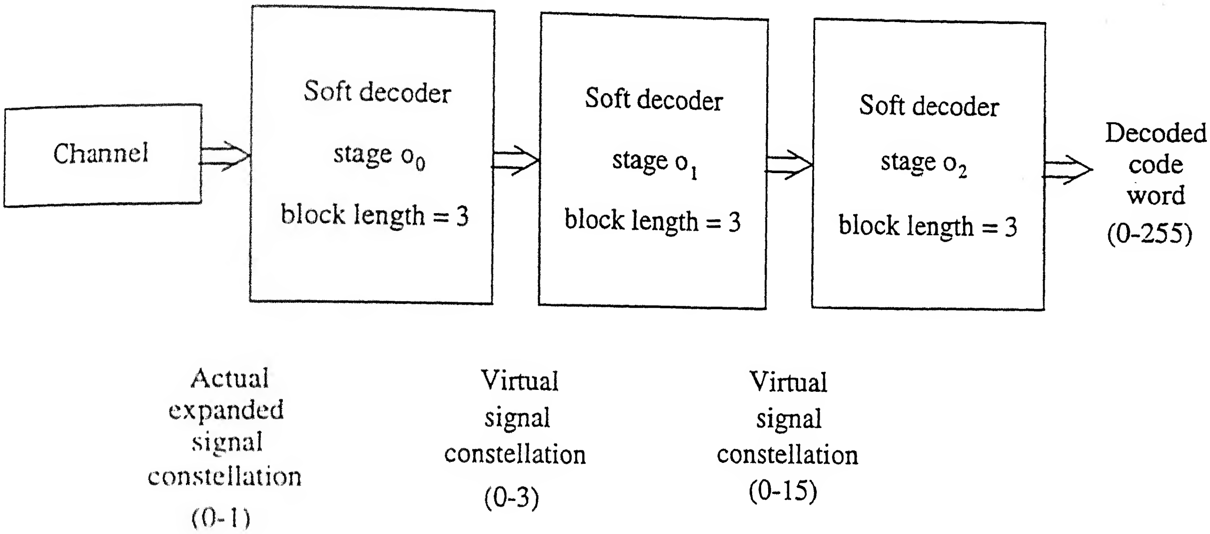

Figure 7.6.2: Soft decoder for concatenated BCM-BCM scheme of Example 7.6.1

The soft decoder at stage o_1, will have to perform soft decoding 3 times of the symbols received from stage o_0 of length 3 and the maximum number of states at this stage can be 16.

The soft decoder at stage 0_2, will have to perform soft decoding once of the symbols received from stage o_1 of length 3 and the maximum number of states at this stage can be 256.

This is used, instead of a single soft decoder of length 27 and having at most 256 states, for the concatenated scheme.

7.7 Examples

Appendix B, gives examples of codes found by a computer search. The codes consists of concatenation of two BCM schemes of block length 3. The example uses an 8-PSK signal constellation as the actual expanded channel signal constellation. A virtual channel signal constellation consisting of 44-point BCM scheme and 64-point BCM scheme is used in the concatenated BCM-BCM scheme. Codes are also obtained for a 7-PSK actual expanded channel signal constellation and a virtual channel signal constellation of 57-points.

7.8 Concluding Remarks

In this chapter, a scheme is presented to use block codes of short block lengths to obtain block codes of larger block lengths, using concatenation. Both the inner and the outer codes correspond to a BCM scheme. The inner block code is considered as a virtual expanded channel signal constellation by the outer block code. The scheme considers all block codes of a fixed block length. The code search, encoding and soft decoding complexity of these codes is reduced as compared to directly using a block code of long length.

Chapter 8

Conclusions

The central concern of investigations in this thesis has been with the code search, block encoding and soft decoding for general block codes used in BCM schemes. Major contributions of the thesis are the development of a set-theoretic framework for code search of general (non-linear) block codes and the development of efficient encoding and soft decoding schemes. This chapter summarizes the work presented in this thesis and gives some suggestions for future work.

- A set-theoretic representation for Euclidean distances between signals from an arbitrary channel signal constellation is obtained. A compact representation for the Euclidean distance distribution of sequences of signals of finite length has been described. This general representation results in schemes of coding and decoding which can work with arbitrary signal constellations.

- A new approach to the problem of obtaining codes for BCM, **the structured distance approach**, is proposed. This is suitable for obtaining general codes, over arbitrary channel signal constellations, to be used with BCM schemes. General (non linear) codes have been reported with redundancy in space and time over various channel signal constellations. Some of the expanded channel signal constellations used are asymmetric, PSK signal constellations, QAM signal constellations, containing a number of signals not equal to a prime or a prime power. The structured distance approach can be utilized with this wide variety of signal constellations.

- Analogy between the problem of sphere packings and the problem of obtaining codes for BCM schemes in conjunction with the structured distance approach is used to obtain a new class of codes known as **codes based on selective permutations of distances**. This gives one method for the selection of the distance distribution required in the structured distance approach.

- A reduced tree based soft decoder for block codes is proposed. This has the advantage of parallel implementation. Use of the reduced tree also eliminates the back tracking required in the Viterbi algorithm. Various schemes for the code tree reduction are discussed in detail. The trade offs provided by the reduced tree based decoder and the suitability of the proposed technique for implementation of fast soft decoders are explained.

- The concept of the optimal minimal trellis for a general code, based on the minimal trellises of all the equivalent codes of a given code, is discussed. The optimal minimal proper trellis for a general block code is obtained from the reduced code tree of equivalent codes.

- A scheme for obtaining the optimal minimal trellis, which in general can be improper, for general codes of block length 3, is given.

- Using concatenation of BCM schemes, a scheme is given to obtain general (non-linear) block codes of long length using codes of short block lengths. The code search, block encoding and soft decoding complexity is reduced due to concatenation. The concatenated BCM-BCM scheme reported in the thesis uses general codes of block length 3.

The thesis thus develops a framework and presents schemes for the code search, block encoding and soft decoding of general block codes to be used with BCM schemes. Block encoding and soft decoding of codes which need not be linear, cyclic, lattice, group, GU or rectangular codes are discussed.

8.1 Unfinished Work and Suggestions for Future Work

- The Euclidean spaces discussed in Chapter 2 can be further studied to find the suitability of certain classes of Euclidean spaces for obtaining certain classes of codes.

- Other classes of codes employing the structured distance approach can be found by imposing other restrictions on the selection of distances to obtain suitable distance distribution. A code search under this new criteria can be conducted.

- The results presented in appendix B are not based on any specific search pattern for the selection of distribution. Other codes can be found by using certain specific pattern for searching the distances to obtain the distance distribution of block codes.

- The reduced tree based decoder can be practically implemented using parallel schemes and a comparison of tree and trellis based soft decoding in real time can be carried out.

- Effort can be put to find schemes to obtain optimal minimal trellis for block codes of fixed length say $n \geq 4$.

- Analysis can be carried out for various concatenated BCM-BCM schemes based on the frame work proposed in Chapter 7. Suitability of using certain specific class of block codes as the component codes for a concatenated scheme for specific applications can be explored.

References

[1] J.-P. Adoul and M. Barth. Nearest neighbor algorithm for spherical codes from the Leech lattice. *IEEE Trans. Inform. Theory*, 34(5):1188–1202, September 1988.

[2] Franz Aurenhammer. Voronoi diagrams – A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[3] E. Ayanoglu and R. M. Gray. The design of joint source and channel trellis waveform coders. *IEEE Trans. Inform. Theory*, IT-33(6):855–865, November 1987.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, IT-20:284–287, March 1974.

[5] R. F. Baldini and P. G. Farrell. Multilevel block subcodes for coded phase modulation. *Electronics Letters*, 30(12):927–929, June 1994.

[6] R. Baldini, F. and P. G. Farrell. Coded modulation based on rings of integers modulo-q–Part I: Block codes. *IEE Proc. Commun.*, 141(3):129–136, June 1994.

[7] R. Baldini, F. and P. G. Farrell. Coded modulation based on rings of integers modulo-q–Part II: Convolutional codes. *IEE Proc. Commun.*, 141(3):137–142, June 1994.

[8] Y. Be'ery, B. Shahar, and J. Snyders. Fast decoding of the Leech lattice. *IEEE J. Select. Area Commun.*, 7(6):959–967, August 1989.

[9] S. Benedetto, R. Garello, M. Mondin, and G. Montorsi. Geometrically uniform partitions of L×MPSK constellations and related binary trellis codes. *IEEE Trans. Inform. Theory*, 39(6):1773–1798, November 1993.

[10] S. Benedetto, R. Garello, M. Mondin, and G. Montorsi. Geometrically uniform TCM codes over groups based on L×MPSK constellations. *IEEE Trans. Inform. Theory*, 40(1):137–152, January 1994.

[11] S. Benedetto, M. A. Marsan, G. Albertengo, and E. Giachin. Combined coding and modulation: Theory and applications. *IEEE Trans. Inform. Theory*, 34(2):223–236, March 1988.

[12] S. Benedetto, M. Mondin, and G. Montorsi. Performance evaluation of trellis-coded modulation schemes. *Proc. IEEE*, 82(6):833–855, June 1994.

[13] S. Benedetto and G. Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. *IEEE Trans. Inform. Theory*, 42:409–428, March 1996.

[14] Y. Berger and Y. Be'er. Bounds on the trellis size of linear block codes. *IEEE Trans. Inform. Theory*, 39(1):203–209, January 1993.

[15] Ezio Biglieri, Dariush Divsalar, Peter J. McLane, and Marvin K. Simon. *Introduction to Trellis-Coded Modulation with Applications*. Maxwell Macmillan International Editions, 1991.

[16] Richard E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesely Publishing Company, 1983.

[17] R. D. Buda. Some optimal codes have structure. *IEEE J. Select. Area Commun.*, 7(6):893–899, August 1989.

[18] A. G. Burr. Block versus trellis: an introduction to coded modulation. *Electro. and Commun. Engg. Jour.*, 5(4):240–248, August 1993.

[19] G. Caire and E. Biglieri. Linear block codes over cyclic groups. *IEEE Trans. Inform. Theory*, 41(5):1246–1256, September 1995.

[20] A. R. Calderbank and N. J. A. Sloane. New trellis codes based on lattices and cosets. *IEEE Trans. Inform. Theory*, IT-33(2):177–195, March 1987.

[21] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1988.

[22] R. H. Deng and D. J. Costello. High rate concatenated coding systems using bandwidth efficient trellis inner codes. *IEEE Trans. Commun.*, 37(5):420–427, May 1989.

[23] T. R. Fischer, M. W. Marcellin, and M. Wang. Trellis-coded vector quantization. *IEEE Trans. Inform. Theory*, 37(6):1551–1566, November 1991.

[24] G. D. Forney, Jr. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, March 1973.

[25] G. D. Forney, Jr. Coset codes–Part I: Introduction and geometrical classification. *IEEE Trans. Inform. Theory*, 34(5):1123–1151, September 1988.

[26] G. D. Forney, Jr. Coset codes–Part II: Binary lattices and related codes. *IEEE Trans. Inform. Theory*, 34(5):1152–1187, September 1988.

[27] G. D. Forney, Jr. Multidimensional constellations–Part II: Voronoi constellations. *IEEE J. Select. Area Commun.*, 7(6):941–958, August 1989.

[28] G. D. Forney, Jr. Geometrically uniform codes. *IEEE Trans. Inform. Theory*, 37(5):1241–1260, September 1991.

[29] G. D. Forney, Jr. The Dynamics of group codes: State spaces, trellis diagrams, and canonical encoders. *IEEE Trans. Inform. Theory*, 39(9):1491–1513, September 1993.

[30] G. D. Forney, Jr., L. Brown, M. V. Eyuboglu, and J. L. Moran. The V.34 high-speed modem standard. *IEEE Commun. Mag.*, 34(12):28–33, December 1996.

[31] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi. Efficient modulation for band-limited channels. *IEEE J. Select. Area Commun.*, SAC-2(5):632–647, September 1984.

[32] G. D. Forney, Jr. and L.-F. Wei. Multidimensional constellations–Part I: Introduction, figures of merit, and generalized cross constellations. *IEEE J. Select. Area Commun.*, 7(6):877–892, August 1989.

[33] R. Garello and S. Benedetto. Multilevel construction of block and trellis group codes. *IEEE Trans. Inform. Theory*, 41(5):1257–1264, September 1995.

[34] L. Goldberg. High-speed modems: They're not just for data anymore. *Electro. Design*, pages 47–67, August 1994.

[35] P. G. Gulak and T. Kailath. Locally connected VLSI architectures for the Viterbi algorithm. *IEEE J. Select. Area Commun.*, 6(3):527–537, April 1988.

[36] P. G. Gulak and E. Shwedyk. VLSI structures for Viterbi receivers–Part I: General theory and applications. *IEEE J. Select. Area Commun.*, SAC-4(1):142–154, January 1986.

[37] Simon Haykin. *Digital Communications*. John Wiley & Sons, Inc., 1988.

[38] Simon Haykin. *Communication Systems*. Wiley Eastern Limited, 1992.

[39] H. Herzberg, Y. Be'ery, and J. Snyders. Concatenated multilevel block coded modulation. *IEEE Trans. Commun.*, 41(1):41–49, January 1993.

[40] G. B. Horn and F. R. Kschischang. On the intractability of permuting a block code to minimize trellis complexity. *IEEE Trans. Inform. Theory*, 42(6):2042–2048, November 1996.

[41] H. Imai and S. Hirakawa. A new multilevel coding method using error-correcting codes. *IEEE Trans. Inform. Theory*, IT-23(3):371–377, May 1977.

[42] M. Isaksson and L. H. Zetterberg. Block-coded M-PSK modulation over GF(M). *IEEE Trans. Inform. Theory*, 39(2):337–346, March 1993.

[43] A. Jadhav and M. Siddiqi. Reduced-tree-based soft decoding for block-coded modulation. *IEE Proc. Commun.*, 144(2):79–83, April 1997.

[44] A. N. Jadhav and M. U. Siddiqi. Soft decoding for block coded modulation. In *Proc. Nat. Con. Commun.*, pages 47–50, I.I.T.-Bombay, India, February 1996.

[45] S. Hamidreza Jamali and Tho Le-Ngoc. *Coded Modulation Techniques for Fading Channels*. Kluwer Academic Publishers, 1994.

[46] T. Kasami, T. Takata, T. Fujiwara, and S. Lin. On multilevel block modulation codes. *IEEE Trans. Inform. Theory*, 37(4):965–975, July 1991.

[47] T. Kasami, T. Takata, T. Fujiwara, and S. Lin. On complexity of trellis structure of linear block codes. *IEEE Trans. Inform. Theory*, 39(3):1057–1064, May 1993.

[48] T. Kasami, T. Takata, T. Fujiwara, and S. Lin. On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes. *IEEE Trans. Inform. Theory*, 39(1):242–245, January 1993.

[49] F. R. Kschischang. The trellis structure of maximal fixed-cost codes. *IEEE Trans. Inform. Theory*, 42(6):1828–1838, November 1996.

[50] F. R. Kschischang, P. G. D. Buda, and S. Pasupathy. Block coset codes for M-ary phase shift keying. *IEEE J. Select. Area Commun.*, 7(6):900–913, August 1989.

[51] F. R. Kschischang and V. Sorokine. On the Trellis structure of block codes. *IEEE Trans. Inform. Theory*, 41(6):1924–1937, November 1995.

[52] A. Lafourcade and A. Vardy. Lower bounds on trellis complexity of block codes. *IEEE Trans. Inform. Theory*, 41(6):1938–1954, November 1995.

[53] G. R. Lang and F. M. Longstaff. A Leech lattice modem. *IEEE J. Select. Area Commun.*, 7(6):968–973, August 1989.

[54] B. P. Lathi. *Modern Digital and Analog Communication Systems.* Prism Book Pvt. Ltd., 1993.

[55] J. Leech and N. J. A. Sloane. Sphere packings and error-correcting codes. *Can. J. Math.*, XXIII(4):718–745, 1971.

[56] Shu Lin and Jr. Daniel J. Costello. *Error Control Coding - Fundamentals and Applications.* Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632, 1983.

[57] H.-A. Loeliger. Signal sets matched to groups. *IEEE Trans. Inform. Theory*, 37(6):1675–1682, November 1991.

[58] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland Publishing Company, 1977.

[59] J. L. Massey. Coding and modulation in digital communications. In *Proc. 1974 Int. Zurich Seminar on Digital Commun.*, pages E2(1)–(4), Zurich, Switzerland, March 1974.

[60] K. R. Matis and J. W. Modestino. Reduced-search soft-decision trellis decoding of linear block codes. *IEEE Trans. Inform. Theory*, IT-28(2):349–355, March 1982.

[61] D. J. Muder. Minimal trellises for block codes. *IEEE Trans. Inform. Theory*, 34(5):1049–1053, September 1988.

[62] R. Pellizzoni and A. Spalvieri. Binary multilevel coset codes based on Reed-Muller codes. *IEEE Trans. Commun.*, 42(7):2357–2360, July 1994.

[63] P. M. Piret. Algebraic construction of cyclic codes over Z8 with a good Euclidean minimum distance. *IEEE Trans. Inform. Theory*, 41(3):815–818, May 1995.

[64] John G. Proakis. *Digital Communications.* McGraw-Hill Book Company, 1989.

[65] S. Rajpal, D. J. Rhee, and S. Lin. Multidimensional trellis coded phase modulation using a multilevel concatenation approach–Part I: Code design. *IEEE Trans. Commun.*, 45(1):64–72, January 1997.

[66] S. Rajpal, D. J. Rhee, and S. Lin. Multidimensional trellis coded phase modulation using a multilevel concatenation approach–Part II: Codes for the AWGN and fading channels. *IEEE Trans. Commun.*, 45(2):177–186, February 1997.

[67] J. Rifà. Groups of complex integers used as QAM signals. *IEEE Trans. Inform. Theory*, 41(5):1512–1523, September 1995.

[68] C. A. Rogers. The packing of equal spheres. *Proc. London Math. Soc.*, 8:609–620, 1958.

[69] E. J. Rossin, N. T. Sindhushayana, and C. D. Heegard. Trellis group codes for the Gaussian channel. *IEEE Trans. Inform. Theory*, 41(5):1217–1245, September 1995.

[70] S. I. Sayegh. A class of optimum block codes in signal space. *IEEE Trans. Commun.*, COM-34(10):1043–1045, October 1986.

[71] P. Schuurman. A table of state complexity bounds for binary linear codes. *IEEE Trans. Inform. Theory*, 42(6):2034–2042, November 1996.

[72] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3), 1948.

[73] C. E. Shannon. Communication in the presence of noise. *Proc. IRE*, pages 10–21, January 1949.

[74] C. E. Shannon. Probability of error for optimal codes in a Gaussian channel. *Bell Syst. Tech. J.*, 38(3):611–656, 1959.

[75] D. Slepian. Permutation modulation. *Proc. IEEE*, 53:228–236, March 1965.

[76] D. Slepian. Group codes for the Gaussian channel. *Bell Syst. Tech. J.*, 47:575–602, 1968.

[77] N. J. A. Sloane. The packing of spheres. *Scientific American*, 250:116–125, January 1984.

[78] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Trans. Inform. Theory*, IT-28(1):55–67, January 1982.

[79] G. Ungerboeck. Trellis-coded modulation with redundant signal sets–Part I: Introduction. *IEEE Commun. Mag.*, 25(2):5–11, February 1987.

[80] G. Ungerboeck. Trellis-coded modulation with redundant signal sets–Part II: State of the art. *IEEE Commun. Mag.*, 25(2):12–21, February 1987.

[81] V. A. Vaishampayan and N. Farvardin. Joint design of block source codes and modulation signal set. *IEEE Trans. Inform. Theory*, 38(4):1230–1248, July 1992.

[82] A. Vardy and F. R. Kschischang. Proof of a conjecture of McEliece regarding the expansion index of the minimal trellis. *IEEE Trans. Inform. Theory*, 42(6):2027–2034, November 1996.

[83] V. V. Vazirani, H. Saran, and B. Sundar Rajan. An efficient algorithm for constructing minimal trellises for codes over finite abelian groups. *IEEE Trans. Inform. Theory*, 42(6):1839–1854, November 1996.

[84] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decooding algorithm. *IEEE Trans. Inform. Theory*, IT-13:260–269, April 1967.

[85] A. J. Viterbi, J. K. Wolf, E. Zehavi, and R. Padovani. A pragmatic approach to trellis-coded modulation. *IEEE Commun. Mag.*, 27(7):11–19, July 1989.

[86] Andrew J. Viterbi and Jim K. Omura. *Principles of Digital Communication and Coding*. McGraw-Hill Book Company, 1979.

[87] H. S. Wang and N. Moayeri. Trellis coded vector quantization. *IEEE Trans. Commun.*, 40(8):1273–1281, August 1992.

[88] L.-F. Wei. Trellis-coded modulation with multidimensional constellations. *IEEE Trans. Inform. Theory*, IT-33(4):483–501, July 1987.

[89] L.-F. Wei. Rotationally invariant trellis-coded modulations with multidimensional M-PSK. *IEEE J. Select. Area Commun.*, 7(9):1281–1295, December 1989.

[90] J. K. Wolf. Efficient maximum likelihood decoding of linear block codes using a trellis. *IEEE Trans. Inform. Theory*, IT-24(1):76–80, January 1978.

[91] John M. Wozencraft and Irwin Mark Jacobs. *Principles of Communication Engineering*. John Wiley & Sons, Inc., 1967.

[92] E. Zehavi and J. K. Wolf. On the performance evaluation of trellis codes. *IEEE Trans. Inform. Theory*, IT-33(2):196–202, March 1987.

Appendix A

Examples of Multilevel Channel Signal Constellations

This appendix gives a listing of the various channel signal constellations used in the thesis. The matrices of the square of the Euclidean distances between signals of the signal constellation as described in Chapter 2 are also given.

3-PSK Signal Constellation

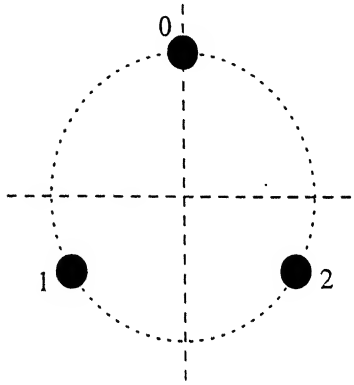

Figure A.2: 3-PSK signal constellation

Table A.2: Euclidean distance matrix for 3-PSK signal constellation

	0	1	2
0	0.0	3.0	3.0
1	3.0	0.0	3.0
2	3.0	3.0	0.0

Binary Signal Constellation

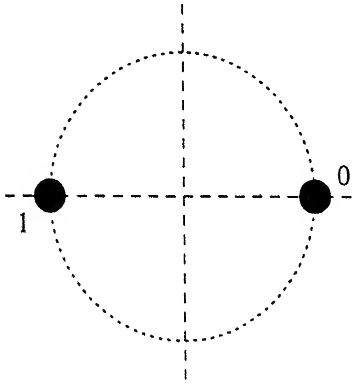

Figure A.1: Binary signal constellation

Table A.1: Euclidean distance matrix for binary signal constellation

	0	1
0	0.0	4.0
1	4.0	0.0

4-PSK Signal Constellation

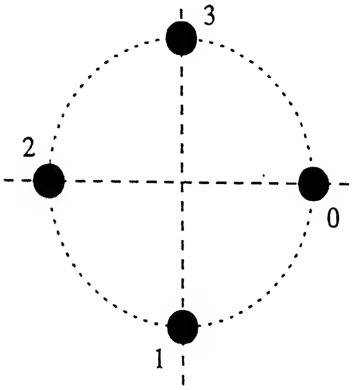

Figure A.3: 4-PSK signal constellation

Table A.3: Euclidean distance matrix for 4-PSK signal constellation

	0	1	2	3
0	0.0	2.0	4.0	2.0
1	2.0	0.0	2.0	4.0
2	4.0	2.0	0.0	2.0
3	2.0	4.0	2.0	0.0

Asymmetric 4-PSK Signal Constellation

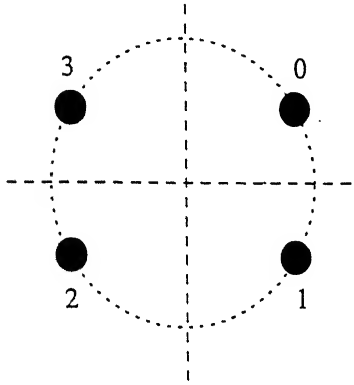

Figure A.4: Asymmetric 4-PSK signal constellation

Table A.4: Euclidean distance matrix for Asymmetric 4-PSK signal constellation

	0	1	2	3
0	0.0	1.0	4.0	3.0
1	1.0	0.0	3.0	4.0
2	4.0	3.0	0.0	1.0
3	3.0	4.0	1.0	0.0

5-PSK Signal Constellation

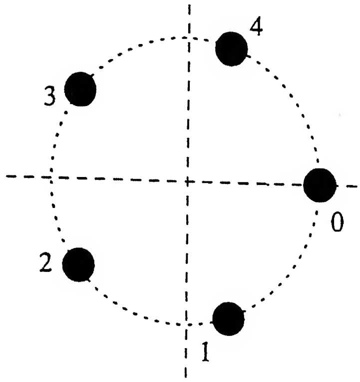

Figure A.5: 5-PSK signal constellation

Table A.5: Euclidean distance matrix for 5-PSK signal constellation

	0	1	2	3	4
0	0.00	1.38	3.62	3.62	1.38
1	1.38	0.00	1.38	3.62	3.62
2	3.62	1.38	0.00	1.38	3.62
3	3.62	3.62	1.38	0.00	1.38
4	1.38	3.62	3.62	1.38	0.00

7-PSK Signal Constellation

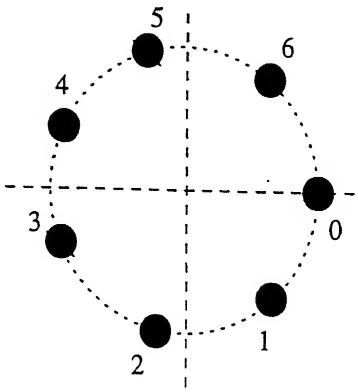

Figure A.7: 7-PSK signal constellation

Table A.7: Euclidean distance matrix for 7-PSK signal constellation

	0	1	2	3	4	5	6
0	0.00	0.75	2.45	3.80	3.80	2.45	0.75
1	0.75	0.00	0.75	2.45	3.80	3.80	2.45
2	2.45	0.75	0.00	0.75	2.45	3.80	3.80
3	3.80	2.45	0.75	0.00	0.75	2.45	3.80
4	3.80	3.80	2.45	0.75	0.00	0.75	2.45
5	2.45	3.80	3.80	2.45	0.75	0.00	0.75
6	0.75	2.45	3.80	3.80	2.45	0.75	0.00

8-PSK Signal Constellation

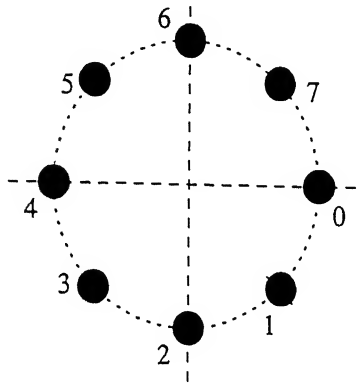

Figure A.8: 8-PSK signal constellation

Table A.8: Euclidean distance matrix for 8-PSK signal constellation

	0	1	2	3	4	5	6	7
0	0.00	0.59	2.00	3.41	4.00	3.41	2.00	0.59
1	0.59	0.00	0.59	2.00	3.41	4.00	3.41	2.00
2	2.00	0.59	0.00	0.59	2.00	3.41	4.00	3.41
3	3.41	2.00	0.59	0.00	0.59	2.00	3.41	4.00
4	4.00	3.41	2.00	0.59	0.00	0.59	2.00	3.41
5	3.41	4.00	3.41	2.00	0.59	0.00	0.59	2.00
6	2.00	3.41	4.00	3.41	2.00	0.59	0.00	0.59
7	0.59	2.00	3.41	4.00	3.41	2.00	0.59	0.00

Asymmetric 8-PSK Signal Constellation

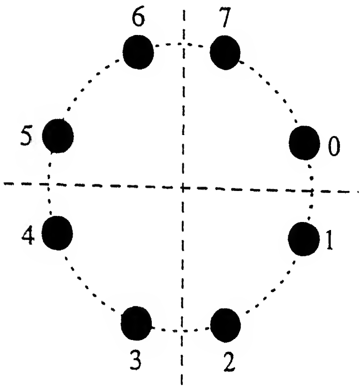

Figure A.9: Asymmetric 8-PSK signal constellation

Table A.9: Euclidean distance matrix for Asymmetric 8-PSK signal constellation

	0	1	2	3	4	5	6	7
0	0.00	0.27	2.00	3.00	4.00	3.73	2.00	1.00
1	0.27	0.00	1.00	2.00	3.73	4.00	3.00	2.00
2	2.00	1.00	0.00	0.27	2.00	3.00	4.00	3.73
3	3.00	2.00	0.27	0.00	1.00	2.00	3.73	4.00
4	4.00	3.73	2.00	1.00	0.00	0.27	2.00	3.00
5	3.73	4.00	3.00	2.00	0.27	0.00	1.00	2.00
6	2.00	3.00	4.00	3.73	2.00	1.00	0.00	0.27
7	1.00	2.00	3.73	4.00	3.00	2.00	0.27	0.00

8-AM PM Signal Constellation

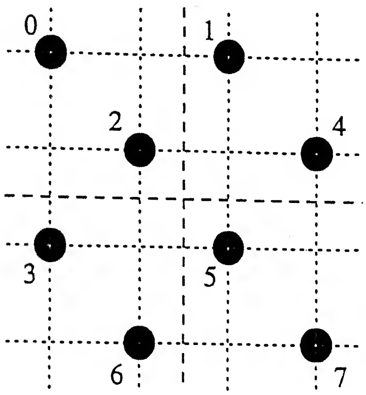

Figure A.10: 8-AM PM signal constellation

Table A.10: Euclidean distance matrix for 8-AM PM signal constellation

	0	1	2	3	4	5	6	7
0	0.00	0.88	0.44	0.88	2.20	1.77	2.20	4.00
1	0.88	0.00	0.44	1.77	0.44	0.88	2.20	2.20
2	0.44	0.44	0.00	0.44	0.88	0.44	0.88	1.77
3	0.88	1.77	0.44	0.00	2.20	0.88	0.44	2.20
4	2.20	0.44	0.88	2.20	0.00	0.44	1.77	0.88
5	1.77	0.88	0.44	0.88	0.44	0.00	0.44	0.44
6	2.20	2.20	0.88	0.44	1.77	0.44	0.00	0.88
7	4.00	2.20	1.77	2.20	0.88	0.44	0.88	0.00

16-PSK Signal Constellation

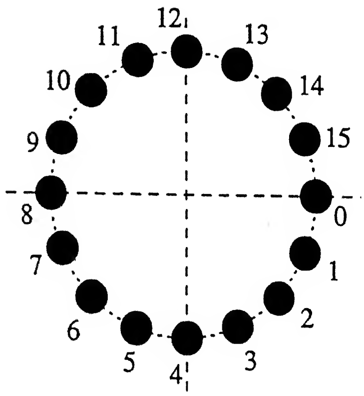

Figure A.11: 16-PSK signal constellation

16-QAM Signal Constellation

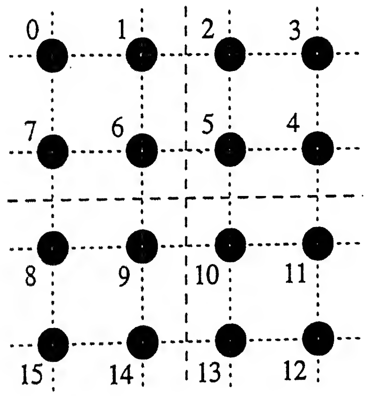

Figure A.12: 16-QAM signal constellation

Table A.11: Euclidean distance matrix for 16-PSK signal constellation

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15
1	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59
2	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23
3	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00
4	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76
5	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41
6	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85
7	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00
8	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41	3.85
9	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76	3.41
10	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00	2.76
11	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23	2.00
12	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59	1.23
13	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15	0.59
14	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00	0.15
15	0.15	0.59	1.23	2.00	2.76	3.41	3.85	4.00	3.85	3.41	2.76	2.00	1.23	0.59	0.15	0.00

Table A.12: Euclidean distance matrix for 16-QAM signal constellation

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0.00	0.22	0.88	1.98	2.20	1.10	0.44	0.22	0.88	1.10	1.77	2.86	4.00	2.86	2.20	1.98
1	0.22	0.00	0.22	0.88	1.10	0.44	0.22	0.44	1.10	0.88	1.10	1.77	2.86	2.20	1.98	2.20
2	0.88	0.22	0.00	0.22	0.44	0.22	0.44	1.10	1.77	1.10	0.88	1.10	2.20	1.98	2.20	2.86
3	1.98	0.88	0.22	0.00	0.22	0.44	1.10	2.20	2.86	1.77	1.10	0.88	1.98	2.20	2.86	4.00
4	2.20	1.10	0.44	0.22	0.00	0.22	0.88	1.98	2.20	1.10	0.44	0.22	0.88	1.10	1.77	2.86
5	1.10	0.44	0.22	0.44	0.22	0.00	0.22	0.88	1.10	0.44	0.22	0.44	1.10	0.88	1.10	1.77
6	0.44	0.22	0.44	1.10	0.88	0.22	0.00	0.22	0.44	0.22	0.44	1.10	1.77	1.10	0.88	1.10
7	0.22	0.44	1.10	2.20	1.98	0.88	0.22	0.00	0.22	0.44	1.10	2.20	2.86	1.77	1.10	0.88
8	0.88	1.10	1.77	2.86	2.20	1.10	0.44	0.22	0.00	0.22	0.88	1.98	2.20	1.10	0.44	0.22
9	1.10	0.88	1.10	1.77	1.10	0.44	0.22	0.44	0.22	0.00	0.22	0.88	1.10	0.44	0.22	0.44
10	1.77	1.10	0.88	1.10	0.44	0.22	0.44	1.10	0.88	0.22	0.00	0.22	0.44	0.22	0.44	1.10
11	2.86	1.77	1.10	0.88	0.22	0.44	1.10	2.20	1.98	0.88	0.22	0.00	0.22	0.44	1.10	2.20
12	4.00	2.86	2.20	1.98	0.88	1.10	1.77	2.86	2.20	1.10	0.44	0.22	0.00	0.22	0.88	1.98
13	2.86	2.20	1.98	2.20	1.10	0.88	1.10	1.77	1.10	0.44	0.22	0.44	0.22	0.00	0.22	0.88
14	2.20	1.98	2.20	2.86	1.77	1.10	0.88	1.10	0.44	0.22	0.44	1.10	0.88	0.22	0.00	0.22
15	1.98	2.20	2.86	4.00	2.86	1.77	1.10	0.88	0.22	0.44	1.10	2.20	1.98	0.88	0.22	0.00

Appendix B

Results of Code Search

B.1 General Block Codes for Arbitrary Channel Signal Constellations

This appendix is a collection of various code listings and a tabulation of some significant illustrative comparisons. The codes listed are general (non-linear) block codes for various expanded channel signal constellations with the Euclidean distance metric which can be used for block coded modulation [BCM] schemes. The particular cases chosen are not application specific, but just to demonstrate the generalizations emphasized in the thesis. This trade-off between redundancy in space and time for coding is manifested in these listings.

The tables of **Block codes for arbitrary channel signal constellations** give results for general (non-linear) codes obtained using an initial code word which is zero and no other optimizations. As explained in Chapter 2 and Chapter 3, depending on the set of Euclidean distances provided by the signals of an expanded channel signal constellation various $d^2_{\min}$ are possible for a BCM scheme. These are tabulated under the column labeled $d^2_{\min}$ and are used for obtaining code words. The range of distances tabulated is selected so that a proper comparison results. The search gives code words such that the Euclidean distance between the code words (that is, the elements in the distance distribution for the code) is greater than or equal to $d^2_{\min}$. The number of the general (non-linear) code words obtained is tabulated under **No. of cd wds**. Comparisons are made with uncoded schemes to obtain asymptotic coding gain **CG dB**, where $CG = 10 \log(d^2_{\min}/d^2_{uncoded})$. The **rate** is equal to

k/n, where $k = \log_Q$(No. of code words), where Q is the number of signals in the uncoded channel signal constellation. Code words are obtained and relevant comparisons are made for a particular channel signal constellation, for various block lengths n.

The entry under the column rate, when equals one, signifies no bandwidth expansion. This corresponds to coded modulation with redundancy in space. The entries less than unit rate, correspond to redundancy in time and space. When the rate is greater than one, the expanded channel signal constellation provides redundancy in space and also an increase in the rate of transmission over the uncoded case. Different codes give different performance and depending on a specific application requirement the proper code can be selected. The codes have been obtained by using a specific search pattern, a different search algorithm will result in different codes. The codes also do not belong to any specific class, but are just based on the structured distance approach philosophy.

Table B.1: Signal constellation is 3PSK and Block length $n = 3$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	6.0	7	0.94	1.77
2	9.0	3	0.53	3.52

Table B.2: Signal constellation is 3PSK and Block length $n = 4$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	6.0	21	1.10	1.77
2	9.0	9	0.79	3.52

Table B.7: Signal constellation is 4PSK and Block length $n = 3$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	32	1.66	0.00
2	6.0	8	1.00	1.77
3	8.0	4	0.66	3.01

Table B.8: Signal constellation is 4PSK and Block length $n = 4$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	128	1.75	0.00
2	6.0	16	1.00	1.77
3	8.0	16	1.00	3.01
4	10.0	4	0.50	3.97

Table B.9: Signal constellation is 4PSK and Block length $n = 5$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	512	1.80	0.00
2	6.0	64	1.20	1.77
3	8.0	32	1.00	3.01
4	10.0	8	0.60	3.97

Table B.10: Signal constellation is 4PSK and Block length $n = 6$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	2048	1.83	0.00
2	6.0	256	1.33	1.77
3	8.0	128	1.16	3.01
4	10.0	16	0.66	3.97

Table B.11: Signal constellation is 4PSK and Block length $n = 7$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	8192	1.86	0.00
2	6.0	1024	1.43	1.77
3	8.0	512	1.29	3.01
4	10.0	64	0.86	3.97

Table B.12: Signal constellation is 4PSK and Block length $n = 8$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	32768	1.86	0.00
2	6.0	2048	1.38	1.77
3	8.0	2048	1.38	3.01
4	10.0	256	1.00	3.97
5	12.0	128	0.88	4.77

Table B.13: Signal constellation is Asymmetric-4PSK and Block length $n = 3$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	8	1.00	0.00
2	5.0	8	1.00	0.97
3	8.0	4	0.66	3.01
4	9.0	2	0.33	3.52

Table B.14: Signal constellation is Asymmetric-4PSK and Block length $n = 4$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	32	1.25	0.00
2	5.0	16	1.00	0.97
3	6.0	16	1.00	1.77
4	8.0	8	0.75	3.01
5	9.0	4	0.50	3.52

Table B.15: Signal constellation is Asymmetric-4PSK and Block length $n = 5$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	64	1.20	0.00
2	5.0	64	1.20	0.97
3	6.0	32	1.00	1.77
4	7.0	32	1.00	2.43
5	8.0	16	0.80	3.01
6	9.0	8	0.60	3.52

Table B.16: Signal constellation is Asymmetric-4PSK and Block length $n = 6$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	256	1.33	0.00
2	5.0	128	1.16	0.97
3	6.0	64	1.00	1.77
4	7.0	64	1.00	2.43
5	8.0	64	1.00	3.01
6	9.0	16	0.66	3.52
7	10.0	16	0.66	3.97
8	11.0	16	0.66	4.39
9	12.0	8	0.50	4.77

Table B.17: Signal constellation is Asymmetric-4PSK and Block length $n = 7$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB
1	4.0	1024	1.43	0.00
2	5.0	256	1.14	0.97
3	6.0	256	1.14	1.77
4	7.0	128	1.00	2.43
5	8.0	128	1.00	3.01
6	9.0	32	0.71	3.52
7	10.0	32	0.71	3.97
8	11.0	16	0.57	4.39
9	12.0	16	0.57	4.77
10	13.0	16	0.57	5.12
11	16.0	8	0.43	6.02

Table B.18: Signal constellation is Asymmetric-4PSK and Block length $n = 8$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary	
			rate	CG dB
1	4.0	4096	1.50	0.00
2	5.0	1024	1.25	0.97
3	6.0	512	1.13	1.77
4	7.0	256	1.00	2.43
5	8.0	256	1.00	3.01
6	9.0	128	0.88	3.52
7	10.0	64	0.75	3.97
8	11.0	64	0.75	4.39
9	12.0	32	0.63	4.77
10	13.0	32	0.63	5.12
11	14.0	16	0.50	5.44

Table B.19: Signal constellation is 5PSK and Block length $n = 3$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary		Comp. with 3-PSK	
			rate	CG dB	rate	CG dB
1	5.00	10	1.11	0.97	0.70	2.22
2	6.38	7	0.94	2.03	0.59	3.28
3	7.24	4	0.66	2.57	0.42	3.83

Table B.20: Signal constellation is 5PSK and Block length $n = 4$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary		Comp. with 3-PSK	
			rate	CG dB	rate	CG dB
1	4.14	39	1.32	0.15	0.83	1.40
2	5.00	20	1.08	0.97	0.68	2.22
3	5.52	18	1.04	1.40	0.66	2.65
4	6.38	14	0.95	2.03	0.60	3.28
5	7.24	11	0.86	2.57	0.55	3.83

Table B.21: Signal constellation is 5PSK and Block length $n = 5$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.14	169	1.49	0.15	0.93	1.40
2	5.00	78	1.26	0.97	0.79	2.22
3	5.52	67	1.21	1.40	0.77	2.65
4	6.38	41	1.07	2.03	0.68	3.28
5	6.90	36	1.03	2.37	0.66	3.62
6	7.24	24	0.91	2.57	0.58	3.83
7	7.76	21	0.88	2.88	0.55	4.13

Table B.22: Signal constellation is 5PSK and Block length $n = 6$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.14	700	1.58	0.15	0.99	1.40
2	5.00	275	1.35	0.97	0.85	2.22
3	5.52	233	1.31	1.40	0.83	2.65
4	6.38	136	1.18	2.03	0.75	3.28
5	6.90	83	1.06	2.37	0.67	3.62
6	7.24	78	1.05	2.57	0.66	3.83
7	7.76	58	0.98	2.88	0.62	4.13
8	8.28	37	0.87	3.16	0.55	4.41

Table B.23: Signal constellation is 5PSK and Block length $n = 7$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.14	2889	1.64	0.15	1.04	1.40
2	5.00	1089	1.44	0.97	0.91	2.22
3	5.52	879	1.40	1.40	0.88	2.65
4	6.38	415	1.24	2.03	0.78	3.28
5	6.90	259	1.15	2.37	0.72	3.62
6	7.24	184	1.07	2.57	0.68	3.83
7	7.76	176	1.06	2.88	0.67	4.13
8	8.28	107	0.96	3.16	0.61	4.41
9	8.62	88	0.92	3.33	0.58	4.58

Table B.24: Signal constellation is 5PSK and Block length $n = 8$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.14	12188	1.70	0.15	1.07	1.40
2	5.00	4267	1.51	0.97	0.95	2.22
3	5.52	3505	1.47	1.40	0.93	2.65
4	6.38	1334	1.30	2.03	0.82	3.28
5	6.90	841	1.21	2.37	0.77	3.62
6	7.24	537	1.13	2.57	0.72	3.83
7	7.76	516	1.12	2.88	0.71	4.13
8	8.28	324	1.04	3.16	0.66	4.41
9	8.62	254	1.00	3.33	0.63	4.58
10	9.13	233	0.98	3.58	0.62	4.83
11	9.14	179	0.94	3.59	0.59	4.84

Table B.25: Signal constellation is 6PSK and Block length $n = 3$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary		Comp. with 3-PSK	
			rate	CG dB	rate	CG dB
1	4.00	14	1.27	0.00	0.80	1.25
2	5.00	12	1.19	0.97	0.75	2.22
3	6.00	7	0.94	1.77	0.59	3.01
4	7.00	4	0.66	2.43	0.42	3.68

Table B.26: Signal constellation is 6PSK and Block length $n = 4$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary		Comp. with 3-PSK	
			rate	CG dB	rate	CG dB
1	4.00	48	1.40	0.00	0.88	1.25
2	5.00	42	1.35	0.97	0.85	2.22
3	6.00	31	1.24	1.77	0.78	3.01
4	7.00	12	0.90	2.43	0.57	3.68
5	8.00	12	0.90	3.01	0.57	4.26
6	9.00	6	0.65	3.52	0.41	4.77

Table B.27: Signal constellation is 6PSK and Block length $n = 5$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary		Comp. with 3-PSK	
			rate	CG dB	rate	CG dB
1	4.00	208	1.54	0.00	0.97	1.25
2	5.00	127	1.40	0.97	0.88	2.22
3	6.00	76	1.25	1.77	0.79	3.01
4	7.00	33	1.01	2.43	0.64	3.68
5	8.00	28	0.96	3.01	0.61	4.26
6	9.00	15	0.78	3.52	0.49	4.77

Table B.28: Signal constellation is 6PSK and Block length $n = 6$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.00	866	1.63	0.00	1.03	1.25
2	5.00	535	1.51	0.97	0.95	2.22
3	6.00	323	1.39	1.77	0.88	3.01
4	7.00	108	1.13	2.43	0.71	3.68
5	8.00	76	1.04	3.01	0.66	4.26
6	9.00	43	0.90	3.52	0.57	4.77
7	10.00	21	0.73	3.97	0.46	5.23

Table B.29: Signal constellation is 6PSK and Block length $n = 7$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.00	4036	1.71	0.00	1.08	1.25
2	5.00	1781	1.54	0.97	0.97	2.22
3	6.00	832	1.39	1.77	0.87	3.01
4	7.00	370	1.22	2.43	0.77	3.68
5	8.00	212	1.10	3.01	0.70	4.26
6	9.00	108	0.96	3.52	0.61	4.77
7	10.00	62	0.85	3.97	0.54	5.23

Table B.30: Signal constellation is 6PSK and Block length $n = 8$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with 3-PSK rate	Comp. with 3-PSK CG dB
1	4.00	20602	1.79	0.00	1.13	1.25
2	5.00	6175	1.57	0.97	0.99	2.22
3	6.00	2783	1.43	1.77	0.90	3.01
4	7.00	1218	1.28	2.43	0.81	3.68
5	8.00	592	1.15	3.01	0.73	4.26
6	9.00	304	1.03	3.52	0.65	4.77
7	10.00	174	0.93	3.97	0.59	5.23
8	11.00	106	0.84	4.39	0.53	5.64

Table B.31: Signal constellation is 7PSK and Block length $n = 3$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.25	55	1.93	-2.50	0.96	0.51
2	2.45	27	1.58	-2.13	0.79	0.88
3	3.20	27	1.58	-0.97	0.79	2.04
4	3.80	17	1.36	-0.22	0.68	2.79
5	3.95	17	1.36	-0.01	0.68	2.96
6	4.55	12	1.19	0.56	0.60	3.57
7	4.90	10	1.11	0.88	0.55	3.89
8	5.30	9	1.06	1.22	0.53	4.23
9	5.65	7	0.94	1.50	0.47	4.51
10	6.25	5	0.77	1.94	0.39	4.95

Table B.32: Signal constellation is 7PSK and Block length $n = 4$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.25	249	1.99	-2.50	1.00	0.51
2	2.45	163	1.84	-2.13	0.92	0.88
3	3.00	98	1.65	-1.25	0.83	1.76
4	3.20	81	1.58	-0.97	0.79	2.04
5	3.80	59	1.47	-0.22	0.74	2.79
6	3.95	59	1.47	-0.01	0.74	2.96
7	4.55	41	1.34	0.56	0.67	3.57
8	4.70	34	1.27	0.70	0.64	3.71
9	4.90	29	1.21	0.88	0.61	3.89
10	5.30	25	1.16	1.22	0.58	4.23
11	5.65	21	1.10	1.50	0.55	4.51
12	6.05	16	1.00	1.80	0.50	4.81
13	6.25	14	0.95	1.94	0.48	4.95
14	6.40	15	0.98	2.04	0.45	5.05
15	7.00	11	0.86	2.43	0.43	5.44

Table B.33: Signal constellation is 7PSK and Block length $n = 5$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.25	1340	2.08	-2.50	1.04	0.51
2	2.45	589	1.84	-2.13	0.92	0.88
3	3.00	475	1.78	-1.25	0.85	1.76
4	3.20	265	1.61	-0.97	0.81	2.04
5	3.75	254	1.60	-0.28	0.80	2.73
6	3.80	205	1.53	-0.22	0.76	2.79
7	3.95	208	1.54	-0.01	0.77	2.96
8	4.55	135	1.42	0.56	0.71	3.57
9	4.70	124	1.39	0.70	0.70	3.71
10	4.90	87	1.29	0.88	0.65	3.89
11	5.30	78	1.26	1.22	0.63	4.23
12	5.45	67	1.21	1.34	0.61	4.35
13	5.65	60	1.18	1.50	0.59	4.51
14	6.05	50	1.13	1.80	0.57	4.81
15	6.25	43	1.09	1.94	0.55	4.95
16	6.40	42	1.08	2.04	0.54	5.05
17	6.80	33	1.00	2.30	0.50	5.31
18	7.00	34	1.01	2.43	0.51	5.44
19	7.15	28	0.96	2.52	0.48	5.53
20	7.35	23	0.90	2.64	0.45	5.65

Table B.34: Signal constellation is 7PSK and Block length $n = 6$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.25	7589	2.15	-2.50	1.08	0.51
2	2.45	3400	1.96	-2.13	0.98	0.88
3	3.00	2305	1.86	-1.25	0.93	1.76
4	3.20	1294	1.72	-0.97	0.86	2.04
5	3.75	1093	1.68	-0.28	0.84	2.73
6	3.80	723	1.58	-0.22	0.79	2.79
7	3.95	716	1.58	-0.01	0.79	2.96
8	4.50	501	1.49	0.51	0.75	3.52
9	4.55	471	1.48	0.56	0.74	3.57
10	4.70	453	1.47	0.70	0.73	3.71
11	4.90	293	1.37	0.88	0.69	3.89
12	5.30	276	1.35	1.22	0.68	4.23
13	5.45	246	1.32	1.34	0.66	4.35
14	5.65	184	1.25	1.50	0.63	4.51
15	6.05	163	1.22	1.80	0.61	4.81
16	6.20	141	1.19	1.90	0.60	4.91
17	6.25	137	1.18	1.94	0.59	4.95
18	6.40	127	1.16	2.04	0.58	5.05
19	6.80	97	1.10	2.30	0.55	5.31
20	7.00	93	1.09	2.43	0.54	5.44
21	7.15	86	1.07	2.52	0.53	5.53
22	7.35	67	1.01	2.64	0.51	5.65
23	7.55	67	1.01	2.75	0.51	5.77
24	7.60	67	1.01	2.79	0.51	5.80
25	7.75	64	1.00	2.87	0.50	5.88
26	7.90	49	0.94	2.96	0.47	5.97
27	8.10	47	0.93	3.06	0.46	6.07

Table B.35: Signal constellation is 7PSK and Block length $n = 7$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.25	44445	2.21	-2.50	1.11	0.51
2	2.45	18731	2.03	-2.13	1.02	0.88
3	3.00	12456	1.94	-1.25	0.97	1.76
4	3.20	5000	1.76	-0.97	0.88	2.04
5	3.80	2760	1.63	-0.22	0.82	2.79
6	3.95	2717	1.62	-0.01	0.81	2.96
7	4.50	1952	1.56	0.51	0.78	3.52
8	4.55	1658	1.53	0.56	0.77	3.57
9	4.70	1660	1.52	0.70	0.76	3.71
10	4.90	1036	1.43	0.88	0.72	3.89
11	5.25	1008	1.42	1.17	0.71	4.19
12	5.45	914	1.41	1.34	0.70	4.35
13	5.65	603	1.32	1.50	0.66	4.51
14	6.05	553	1.30	1.80	0.65	4.81
15	6.20	487	1.28	1.90	0.64	4.91
16	6.25	401	1.24	1.94	0.62	4.95
17	6.40	398	1.23	2.04	0.61	5.05
18	6.80	318	1.19	2.30	0.60	5.31
19	6.95	283	1.16	2.40	0.58	5.41
20	7.15	254	1.14	2.52	0.57	5.53
21	7.35	195	1.08	2.64	0.54	5.65
22	7.55	196	1.09	2.75	0.55	5.77
23	7.60	186	1.07	2.79	0.53	5.80
24	7.75	181	1.07	2.87	0.53	5.88
25	7.90	154	1.04	2.96	0.52	5.97
26	8.10	131	1.00	3.06	0.50	6.07
27	8.30	123	0.99	3.17	0.49	6.18

Table B.36: Signal constellation is 7PSK and Block length $n = 8$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary		Comp. with QPSK	
			rate	CG dB	rate	CG dB
1	3.80	11173	1.68	-0.22	0.84	2.79
2	3.95	11057	1.67	-0.01	0.83	2.96
3	4.50	8213	1.63	0.51	0.81	3.52
4	4.55	5903	1.56	0.56	0.78	3.57
5	4.70	5919	1.57	0.70	0.78	3.71
6	4.90	3960	1.49	0.88	0.75	3.89
7	5.25	3863	1.48	1.17	0.74	4.19
8	5.30	3553	1.47	1.22	0.77	4.23
9	5.65	2081	1.38	1.50	0.69	4.51
10	6.00	1975	1.37	1.76	0.68	4.77
11	6.20	1748	1.35	1.90	0.67	4.91
12	6.25	1270	1.29	1.94	0.64	4.95
13	6.40	1259	1.28	2.04	0.64	5.05
14	6.80	1064	1.26	2.30	0.63	5.31
15	6.95	941	1.23	2.40	0.62	5.41
16	7.15	797	1.20	2.52	0.60	5.53
17	7.35	621	1.16	2.64	0.58	5.65
18	7.55	615	1.15	2.75	0.58	5.77
19	7.60	565	1.14	2.79	0.57	5.80
20	7.70	556	1.13	2.84	0.57	5.85
21	7.90	481	1.11	2.96	0.56	5.97
22	8.10	383	1.07	3.06	0.54	6.07
23	8.30	369	1.06	3.17	0.53	6.18
24	8.35	357	1.06	3.20	0.53	6.20

Table B.37: Signal constellation is 8PSK and Block length $n = 3$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.00	64	2.00	-3.01	1.00	0.00
2	2.59	41	1.79	-1.88	0.90	1.12
3	3.18	31	1.66	-0.99	0.83	2.01
4	3.41	19	1.42	-0.69	0.71	2.32
5	4.00	32	1.67	0.00	0.84	3.01
6	4.59	14	1.27	0.59	0.64	3.61
7	5.41	8	1.00	1.31	0.50	4.32
8	6.00	12	1.19	1.76	0.60	4.77
9	6.59	4	0.67	2.17	0.34	5.18
10	6.82	6	0.86	2.32	0.43	5.33
11	7.41	4	0.67	2.68	0.34	5.69

Table B.38: Signal constellation is 8PSK and Block length $n = 4$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.00	512	2.25	-3.01	1.13	0.00
2	2.36	179	1.87	-2.29	0.94	0.72
3	2.59	163	1.84	-1.88	0.92	1.12
4	3.18	125	1.74	-0.99	0.87	2.01
5	3.41	70	1.53	-0.69	0.77	2.32
6	4.00	128	1.75	0.00	0.88	3.01
7	4.59	41	1.34	0.59	0.67	3.61
8	5.18	29	1.21	1.12	0.61	4.13
9	5.41	22	1.11	1.31	0.56	4.32
10	6.00	24	1.15	1.76	0.58	4.77
11	6.59	15	0.98	2.17	0.45	5.18
12	6.82	12	0.90	2.32	0.45	5.33
13	7.18	11	0.86	2.54	0.43	5.55
14	7.41	12	0.90	2.68	0.45	5.69

Table B.39: Signal constellation is 8PSK and Block length $n = 5$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.00	2048	2.20	-3.01	1.10	0.00
2	2.36	968	1.98	-2.29	0.99	0.72
3	2.59	674	1.88	-1.88	0.94	1.12
4	2.95	548	1.82	-1.32	0.91	1.69
5	3.18	501	1.79	-0.99	0.90	2.01
6	3.41	269	1.61	-0.69	0.81	2.32
7	4.00	512	1.80	0.00	0.90	3.01
8	4.36	155	1.46	0.37	0.73	3.38
9	4.59	135	1.42	0.59	0.71	3.61
10	5.18	101	1.33	1.12	0.67	4.13
11	5.41	69	1.22	1.31	0.61	4.32
12	5.77	82	1.27	1.59	0.64	4.60
13	6.00	72	1.23	1.76	0.62	4.77
14	6.36	44	1.09	2.01	0.55	5.02
15	6.59	42	1.08	2.17	0.54	5.18
16	6.82	34	1.02	2.32	0.51	5.33
17	7.18	29	0.97	2.54	0.49	5.55
18	7.41	26	0.94	2.68	0.47	5.69

Table B.40: Signal constellation is 8PSK and Block length $n = 6$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary		Comp. with QPSK	
			rate	CG dB	rate	CG dB
1	2.00	16384	2.33	-3.01	1.17	0.00
2	2.36	5379	2.07	-2.29	1.04	0.72
3	2.59	4088	2.00	-1.88	1.00	1.12
4	2.95	3094	1.93	-1.32	0.97	1.69
5	3.18	1996	1.83	-0.99	0.92	2.01
6	3.41	1158	1.70	-0.69	0.85	2.32
7	3.54	4096	2.00	-0.53	1.00	2.48
8	4.00	2048	1.83	0.00	0.92	3.01
9	4.36	619	1.55	0.37	0.78	3.38
10	4.59	469	1.48	0.59	0.74	3.61
11	4.95	377	1.43	0.93	0.72	3.94
12	5.18	354	1.41	1.12	0.71	4.13
13	5.41	231	1.30	1.31	0.65	4.32
14	5.77	235	1.31	1.59	0.66	4.60
15	6.00	196	1.27	1.76	0.64	4.77
16	6.36	143	1.19	2.01	0.60	5.02
17	6.59	130	1.17	2.17	0.59	5.18
18	6.82	99	1.11	2.32	0.56	5.33
19	6.95	98	1.10	2.40	0.55	5.41
20	7.18	94	1.09	2.54	0.54	5.55
21	7.41	69	1.02	2.68	0.51	5.69
22	7.77	144	1.19	2.88	0.60	5.89
23	8.00	96	1.10	3.01	0.55	6.02
24	8.36	46	0.92	3.20	0.46	6.21
25	8.59	46	0.92	3.31	0.46	6.33
26	8.82	38	0.87	3.43	0.44	6.44

Table B.41: Signal constellation is Asymmetric-8PSK and Block length $n = 3$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.27	46	1.84	-2.46	0.92	0.55
2	3.00	32	1.67	-1.25	0.84	1.76
3	4.00	32	1.67	0.00	0.84	3.01
4	4.27	13	1.23	0.28	0.62	3.29
5	4.54	10	1.11	0.55	0.55	3.56
6	4.73	10	1.11	0.73	0.55	3.74
7	5.00	10	1.11	0.97	0.55	3.98
8	5.27	8	1.00	1.20	0.50	4.21
9	6.00	5	0.77	1.76	0.39	4.77

Table B.42: Signal constellation is Asymmetric-8PSK and Block length $n = 4$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.27	194	1.90	-2.46	0.95	0.55
2	3.00	128	1.75	-1.25	0.88	1.76
3	4.00	128	1.75	0.00	0.88	3.01
4	4.27	47	1.39	0.28	0.69	3.29
5	4.54	36	1.29	0.55	0.65	3.56
6	4.73	32	1.25	0.73	0.63	3.74
7	4.81	28	1.20	0.80	0.60	3.81
8	5.00	27	1.19	0.97	0.59	3.98
9	5.27	29	1.21	1.20	0.61	4.21
10	5.54	22	1.11	1.41	0.56	4.42
11	5.73	14	0.95	1.56	0.48	4.57
12	6.00	16	1.00	1.76	0.50	4.77
13	6.27	12	0.90	1.95	0.45	4.96
14	6.54	11	0.86	2.14	0.43	5.15

Table B.43: Signal constellation is Asymmetric-8PSK and Block length $n = 5$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary		Comp. with QPSK	
			rate	CG dB	rate	CG dB
1	2.27	767	1.92	-2.46	0.96	0.55
2	3.00	512	1.80	-1.25	0.90	1.76
3	4.00	512	1.80	0.00	0.90	3.01
4	4.27	174	1.49	0.28	0.74	3.29
5	4.54	126	1.40	0.55	0.70	3.56
6	4.73	101	1.33	0.73	0.66	3.74
7	4.81	103	1.34	0.80	0.67	3.81
8	5.00	96	1.32	0.97	0.66	3.98
9	5.08	87	1.29	1.04	0.64	4.05
10	5.27	87	1.29	1.20	0.64	4.21
11	5.54	69	1.22	1.41	0.61	4.42
12	5.73	59	1.18	1.56	0.59	4.57
13	5.81	55	1.16	1.62	0.58	4.63
14	6.00	58	1.17	1.76	0.59	4.77
15	6.27	47	1.11	1.95	0.56	4.96
16	6.54	35	1.03	2.14	0.51	5.15
17	6.73	30	0.98	2.26	0.49	5.27
18	7.00	32	1.00	2.43	0.50	5.44
19	7.27	32	1.00	2.59	0.50	5.61
20	7.54	32	1.00	2.75	0.50	5.76
21	8.00	32	1.00	3.01	0.50	6.02
22	8.27	18	0.83	3.15	0.42	6.16
23	8.54	15	0.78	3.29	0.39	6.30

Table B.44: Signal constellation is Asymmetric-8PSK and Block length $n = 6$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.08	4001	1.99	-2.84	0.99	0.17
2	2.27	3295	1.95	-2.46	0.97	0.55
3	2.35	3072	1.93	-2.31	0.96	0.70
4	3.00	2048	1.83	-1.25	0.92	1.76
5	4.00	2048	1.83	0.00	0.92	3.01
6	4.08	633	1.55	0.01	0.78	3.10
7	4.27	620	1.54	0.28	0.77	3.29
8	4.35	483	1.49	0.36	0.74	3.37
9	4.54	478	1.48	0.55	0.74	3.56
10	4.73	363	1.42	0.73	0.71	3.74
11	4.81	362	1.41	0.80	0.70	3.81
12	5.00	308	1.38	0.97	0.69	3.98
13	5.08	298	1.37	1.04	0.68	4.05
14	5.27	284	1.36	1.20	0.67	4.21
15	5.54	226	1.30	1.41	0.65	4.42
16	5.73	188	1.26	1.56	0.63	4.57
17	5.81	188	1.26	1.62	0.63	4.63
18	6.00	176	1.24	1.76	0.62	4.77
19	6.08	147	1.20	1.82	0.60	4.83
20	6.27	149	1.20	1.95	0.60	4.96
21	6.54	120	1.15	2.14	0.58	5.15
22	6.73	103	1.11	2.26	0.56	5.27
23	6.81	96	1.10	2.31	0.55	5.32
24	7.00	97	1.10	2.43	0.55	5.44
25	7.27	96	1.10	2.59	0.55	5.61
26	7.54	128	1.17	2.75	0.58	5.76
27	8.00	128	1.17	3.01	0.58	6.02
28	8.27	47	0.92	3.15	0.46	6.16
29	8.54	38	0.87	3.29	0.44	6.30

Table B.45: Signal constellation is 8 AM-PM and Block length $n = 3$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary		Comp. with QPSK	
			rate	CG dB	rate	CG dB
1	2.20	22	1.49	-2.60	0.74	0.41
2	2.21	16	1.33	-2.58	0.67	0.43
3	2.64	16	1.33	-1.80	0.67	1.21
4	2.65	14	1.27	-1.79	0.64	1.22
5	3.08	12	1.19	-1.14	0.60	1.88
6	3.54	11	1.15	-0.53	0.58	2.47
7	3.96	8	1.00	-0.01	0.50	2.96
8	4.00	8	1.00	0.00	0.50	3.01
9	4.40	6	0.86	0.41	0.43	3.42
10	4.84	5	0.77	0.83	0.39	3.84

Table B.46: Signal constellation is 8 AM-PM and Block length $n = 4$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary		Comp. with QPSK	
			rate	CG dB	rate	CG dB
1	2.20	97	1.65	-2.60	0.82	0.41
2	2.26	64	1.50	-2.48	0.75	0.43
3	2.64	62	1.49	-1.80	0.74	1.21
4	2.65	42	1.35	-1.79	0.67	1.22
5	3.08	41	1.34	-1.14	0.67	1.88
6	3.09	28	1.20	-1.12	0.60	1.89
7	3.52	30	1.23	-0.56	0.61	2.46
8	3.54	29	1.21	-0.53	0.61	2.47
9	3.96	22	1.11	-0.01	0.56	2.96
10	3.97	20	1.08	0.00	0.54	2.97
11	3.98	16	1.00	0.00	0.50	2.98
12	4.00	20	1.08	0.00	0.54	3.01
13	4.40	18	1.04	0.41	0.52	3.42
14	4.41	13	0.93	0.42	0.46	3.43
15	4.42	14	0.95	0.43	0.48	3.45
16	4.84	12	0.90	0.83	0.45	3.84

Table B.47: Signal constellation is 8 AM-PM and Block length $n = 5$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.20	388	1.72	-2.60	0.86	0.41
2	2.21	256	1.60	-2.58	0.80	0.43
3	2.64	242	1.58	-1.80	0.79	1.21
4	2.65	153	1.45	-1.79	0.73	1.22
5	3.08	148	1.44	-1.14	0.72	1.88
6	3.09	105	1.34	-1.12	0.66	1.89
7	3.52	109	1.35	-0.56	0.67	2.46
8	3.53	80	1.26	-0.54	0.63	2.47
9	3.54	82	1.27	-0.53	0.64	2.48
10	3.96	63	1.20	-0.01	0.60	2.96
11	3.97	51	1.13	0.00	0.57	2.97
12	3.98	43	1.09	0.00	0.54	2.99
13	4.00	51	1.13	0.00	0.57	3.01
14	4.40	51	1.13	0.41	0.57	3.42
15	4.41	37	1.04	0.42	0.52	3.43
16	4.42	35	1.03	0.43	0.51	3.45
17	4.44	34	1.02	0.45	0.51	3.46
18	4.84	33	1.01	0.83	0.50	3.84
19	4.85	29	0.97	0.84	0.49	3.85
20	4.86	26	0.94	0.85	0.47	3.86

Table B.48: Signal constellation is 8 AM-PM and Block length $n = 6$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB
1	2.20	1700	1.79	-2.60	0.89	0.41
2	2.21	1096	1.68	-2.58	0.84	0.43
3	2.64	1098	1.68	-1.80	0.84	1.21
4	2.65	573	1.53	-1.79	0.76	1.22
5	3.08	550	1.52	-1.14	0.75	1.88
6	3.09	399	1.44	-1.12	0.72	1.89
7	3.52	412	1.45	-0.56	0.72	2.46
8	3.53	250	1.32	-0.54	0.66	2.47
9	3.54	233	1.31	-0.53	0.65	2.48
10	3.96	218	1.29	-0.01	0.64	2.96
11	3.97	152	1.21	0.00	0.60	2.97
12	3.98	147	1.20	0.00	0.60	2.99
13	4.00	157	1.21	0.00	0.61	3.01
14	4.40	154	1.21	0.41	0.60	3.42
15	4.41	110	1.13	0.42	0.57	3.43
16	4.42	101	1.11	0.43	0.56	3.45
17	4.44	100	1.10	0.45	0.55	3.46
18	4.84	99	1.11	0.83	0.56	3.84
19	4.85	80	1.05	0.84	0.53	3.85
20	4.86	74	1.03	0.85	0.52	3.86
21	4.88	73	1.03	0.86	0.52	3.87
22	5.28	78	1.04	1.20	0.52	4.21
23	5.29	60	0.98	1.21	0.49	4.22
24	5.30	54	0.96	1.22	0.48	4.23
25	5.31	54	0.96	1.23	0.48	4.24
26	5.32	52	0.95	1.24	0.47	4.25

Table B.49: Signal constellation is 16PSK and Block length $n = 3$

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB	Comp. with 8PSK rate	Comp. with 8PSK CG dB
1	0.59	512	3.00	-8.31	1.50	-5.30	1.00	0.00
2	0.74	333	2.79	-7.33	1.40	-4.32	0.93	0.98
3	0.89	230	2.62	-6.53	1.31	-3.52	0.87	1.79
4	1.18	149	2.41	-5.30	1.21	-2.29	0.81	3.01
5	1.23	138	2.37	-5.12	1.19	-2.11	0.79	3.19
6	1.33	160	2.44	-4.78	1.22	-1.77	0.81	3.53
7	1.38	104	2.23	-4.62	1.12	-1.61	0.75	3.69
8	1.53	100	2.21	-4.17	1.11	-1.16	0.74	4.14
9	1.77	100	2.21	-3.54	1.11	-0.53	0.74	4.77
10	1.82	100	2.21	-3.42	1.11	-0.41	0.74	4.89
11	1.97	80	2.11	-3.08	1.06	-0.01	0.71	5.24
12	2.00	80	2.11	-3.01	1.06	0.00	0.71	5.30
13	2.15	56	1.94	-2.70	0.97	0.31	0.65	5.62
14	2.30	48	1.86	-2.40	0.93	0.61	0.62	5.91
15	2.41	47	1.85	-2.20	0.93	0.81	0.62	6.11
16	2.46	45	1.83	-2.11	0.92	0.90	0.61	6.20

Table B.50: Signal constellation is 16PSK and Block length $n = 4$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB	Comp. with 8PSK rate	Comp. with 8PSK CG dB
1	0.59	8192	3.25	-8.31	1.63	-5.30	1.09	0.00
2	0.60	2864	2.87	-8.24	1.44	-5.22	0.96	0.01
3	0.74	2664	2.84	-7.33	1.42	-4.32	0.95	0.98
4	0.89	1867	2.72	-6.53	1.36	-3.52	0.91	1.79
5	1.04	1120	2.53	-5.85	1.27	-2.84	0.85	2.46
6	1.18	917	2.46	-5.30	1.23	-2.29	0.82	3.01
7	1.23	806	2.41	-5.12	1.21	-2.11	0.81	3.19
8	1.33	856	2.44	-4.78	1.22	-1.77	0.81	3.53
9	1.38	752	2.39	-4.62	1.20	-1.61	0.80	3.69
10	1.48	592	2.30	-4.32	1.15	-1.31	0.77	3.99
11	1.53	524	2.26	-4.17	1.13	-1.16	0.75	4.14
12	1.68	488	2.23	-3.77	1.12	-0.76	0.74	4.54
13	1.77	484	2.22	-3.54	1.11	-0.53	0.74	4.77
14	1.82	496	2.24	-3.42	1.12	-0.41	0.75	4.89
15	1.92	464	2.21	-3.19	1.10	-0.18	0.73	5.12
16	1.97	400	2.16	-3.08	1.08	-0.01	0.72	5.24
17	2.00	336	2.10	-3.01	1.05	0.00	0.70	5.30
18	2.12	267	2.02	-2.75	1.01	0.25	0.67	5.55
19	2.15	230	1.96	-2.70	0.98	0.31	0.65	5.62
20	2.30	221	1.95	-2.40	0.97	0.61	0.65	5.91
21	2.36	209	1.92	-2.29	0.96	0.72	0.64	6.02
22	2.41	211	1.93	-2.20	0.96	0.81	0.64	6.11
23	2.45	196	1.90	-2.13	0.95	0.88	0.63	6.18
24	2.46	196	1.90	-2.11	0.95	0.90	0.63	6.20
25	2.56	189	1.89	-1.94	0.94	1.07	0.62	6.37

Table B.51: Signal constellation is 16PSK and Block length $n = 5$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary		Comp. with QPSK		Comp. with 8PSK	
			rate	CG dB	rate	CG dB	rate	CG dB
1	0.59	65536	3.20	-8.31	1.60	-5.30	1.07	0.00
2	0.74	21197	2.87	-7.33	1.44	-4.32	0.96	0.98
3	0.89	14991	2.77	-6.53	1.39	-3.52	0.93	1.79
4	1.18	6223	2.52	-5.30	1.26	-2.29	0.84	3.01
5	1.23	4940	2.45	-5.12	1.23	-2.11	0.82	3.19
6	1.33	4532	2.43	-4.78	1.22	-1.77	0.81	3.53
7	1.38	4532	2.43	-4.62	1.22	-1.61	0.81	3.69
8	1.53	3052	2.32	-4.17	1.16	-1.16	0.77	4.14
9	1.77	2508	2.26	-3.54	1.13	-0.53	0.75	4.77
10	1.83	2464	2.25	-3.40	1.12	-0.39	0.74	4.91
11	2.00	1808	2.16	-3.01	1.08	0.00	0.72	5.30
12	2.07	1495	2.11	-2.86	1.05	0.15	0.70	5.45
13	2.15	1139	2.03	-2.70	1.02	0.31	0.66	5.62
14	2.27	1139	2.03	-2.46	1.02	0.55	0.66	5.85
15	2.36	992	1.99	-2.29	1.00	0.72	0.66	6.02
16	2.41	980	1.98	-2.20	0.99	0.81	0.66	6.11
17	2.45	928	1.97	-2.13	0.98	0.88	0.65	6.18
18	2.46	906	1.96	-2.11	0.98	0.90	0.65	6.20
19	2.51	901	1.96	-1.94	0.98	1.07	0.65	6.37

Table B.52: Signal constellation is 16QAM and Block length $n = 3$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB	Comp. with 8PSK rate	Comp. with 8PSK CG dB
1	0.44	2048	3.66	-9.59	1.83	-6.58	1.22	-1.27
2	0.66	512	3.00	-7.83	1.50	-4.81	1.00	0.49
3	0.88	256	2.67	-6.58	1.34	-3.57	0.89	1.74
4	1.10	105	2.24	-5.61	1.12	-2.60	0.75	2.71
5	1.32	74	2.07	-4.81	1.04	-1.80	0.69	3.50
6	1.54	63	1.99	-4.15	0.99	-1.14	0.65	4.17
7	1.76	64	2.00	-3.57	1.00	-0.56	0.66	4.75
8	1.98	64	2.00	-3.05	1.00	-0.01	0.66	5.26
9	1.99	36	1.72	-3.03	0.86	0.00	0.57	5.28
10	2.20	37	1.74	-2.60	0.87	0.41	0.58	5.72
11	2.21	31	1.65	-2.58	0.83	0.43	0.55	5.74
12	2.42	31	1.65	-2.18	0.83	0.83	0.55	6.13
13	2.86	32	1.67	-1.46	0.84	1.55	0.56	6.86
14	3.96	32	1.67	-0.01	0.84	2.97	0.56	8.27
15	4.00	16	1.33	0.00	0.67	3.01	0.45	8.31
16	4.18	11	1.15	0.19	0.58	3.20	0.39	8.50
17	4.22	10	1.11	0.23	0.56	3.24	0.37	8.54
18	4.40	9	1.01	0.41	0.50	3.42	0.33	8.73
19	4.44	7	0.94	0.45	0.47	3.46	0.31	8.77
20	4.62	7	0.94	0.63	0.47	3.64	0.31	8.94
21	4.63	7	0.94	0.64	0.47	3.65	0.31	8.95
22	4.84	8	1.00	0.83	0.50	3.84	0.33	9.14
23	4.88	7	0.94	0.86	0.47	3.87	0.31	9.18
24	5.06	8	1.00	1.02	0.50	4.03	0.33	9.33
25	5.10	6	0.86	1.06	0.43	4.07	0.29	9.37
26	5.94	8	1.00	1.71	0.50	4.72	0.33	10.03
27	5.98	6	0.86	1.75	0.43	4.76	0.29	10.06
28	6.20	4	0.67	1.90	0.34	4.91	0.23	10.22
29	6.86	4	0.67	2.34	0.34	5.35	0.23	10.65

Table B.53: Signal constellation is 16QAM and Block length $n = 4$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	Comp. with binary CG dB	Comp. with QPSK rate	Comp. with QPSK CG dB	Comp. with 8PSK rate	Comp. with 8PSK CG dB
1	0.88	4096	3.00	-6.58	1.50	-3.57	1.00	1.74
2	1.10	619	2.32	-5.61	1.16	-2.60	0.77	2.71
3	1.32	439	2.19	-4.81	1.10	-1.80	0.73	3.50
4	1.54	281	2.03	-4.15	1.02	-1.14	0.68	4.17
5	1.76	250	1.99	-3.57	1.00	-0.56	0.66	4.75
6	1.77	233	1.97	-3.54	0.99	-0.53	0.66	4.77
7	1.98	231	1.96	-3.05	0.98	-0.01	0.65	5.26
8	1.99	132	1.76	-3.03	0.88	0.00	0.59	5.28
9	2.20	128	1.75	-2.60	0.87	0.41	0.58	5.72
10	2.21	114	1.71	-2.58	0.86	0.43	0.57	5.74
11	2.42	112	1.70	-2.18	0.85	0.83	0.56	6.13
12	2.64	126	1.74	-1.80	0.87	1.21	0.58	6.51
13	2.86	128	1.75	-1.46	0.87	1.55	0.58	6.86
14	3.08	128	1.75	-1.14	0.87	1.88	0.58	7.18
15	3.96	128	1.75	-0.01	0.87	2.97	0.58	8.27
16	3.97	40	1.33	0.00	0.67	2.98	0.45	8.28
17	4.00	40	1.33	0.00	0.67	3.01	0.45	8.31
18	4.18	39	1.32	0.19	0.66	3.20	0.44	8.50
19	4.19	30	1.23	0.20	0.61	3.21	0.41	8.51
20	4.22	31	1.24	0.23	0.62	3.24	0.41	8.54
21	4.40	27	1.19	0.41	0.60	3.42	0.40	8.73
22	4.42	24	1.15	0.43	0.58	3.44	0.39	8.75
23	4.44	23	1.13	0.45	0.57	3.46	0.38	8.77
24	4.62	23	1.13	0.63	0.57	3.64	0.38	8.94
25	4.63	19	1.06	0.64	0.53	3.65	0.35	8.95
26	4.65	20	1.08	0.65	0.54	3.66	0.36	8.97
27	4.84	19	1.06	0.83	0.53	3.84	0.35	9.14
28	4.88	17	1.02	0.86	0.51	3.87	0.34	9.18
29	5.06	17	1.02	1.02	0.51	4.03	0.34	9.33
30	5.10	16	1.00	1.06	0.50	4.07	0.33	9.37
31	5.94	16	1.00	1.72	0.50	4.72	0.33	10.03
32	5.95	15	0.98	1.73	0.49	4.73	0.32	10.04
33	5.98	16	1.00	1.75	0.50	4.76	0.33	10.06
34	6.16	16	1.00	1.88	0.50	4.89	0.33	10.19
35	6.20	12	0.90	1.90	0.45	4.91	0.30	10.22
36	6.38	16	1.00	2.03	0.50	5.04	0.33	10.34
37	6.82	16	1.00	2.32	0.50	5.33	0.33	10.63
38	6.86	12	0.90	2.34	0.45	5.35	0.30	10.65
39	7.92	16	1.00	2.97	0.50	5.98	0.33	11.28
40	8.00	8	0.75	3.01	0.38	6.02	0.25	11.32
41	8.18	5	0.58	3.11	0.29	6.12	0.19	11.42
42	8.21	4	0.50	3.12	0.25	6.13	0.17	11.43

Table B.54: Signal constellation is 16QAM and Block length $n = 5$

Sr. No.	d^2_{min}	No. of cd wds	Comp. with binary rate	CG dB	Comp. with QPSK rate	CG dB	Comp. with 8PSK rate	CG dB
1	0.88	32768	3.00	-6.58	1.50	-3.57	1.00	1.73
2	1.10	4561	2.43	-5.61	1.22	-2.60	0.81	1.74
3	1.32	1919	2.18	-4.81	1.09	-1.80	0.73	3.50
4	1.54	1484	2.11	-4.15	1.05	-1.14	0.70	4.17
5	1.76	1158	2.04	-3.57	1.02	-0.56	0.67	4.75
6	1.98	763	1.92	-3.05	0.96	-0.01	0.64	5.26
7	2.20	521	1.81	-2.60	0.90	0.41	0.60	5.72
8	2.21	432	1.75	-2.58	0.88	0.43	0.58	5.75
9	2.42	419	1.74	-2.18	0.87	0.83	0.58	6.13
10	2.64	419	1.74	-1.80	0.87	1.21	0.58	6.51
11	2.86	486	1.78	-1.46	0.89	1.55	0.59	6.86
12	3.08	512	1.80	-1.14	0.90	1.88	0.60	7.18
13	3.09	512	1.80	-1.12	0.90	1.89	0.60	7.19
14	3.30	512	1.80	-0.84	0.90	2.17	0.60	7.48
15	3.52	512	1.80	-0.55	0.90	2.46	0.60	7.76
16	3.96	512	1.80	-0.01	0.90	2.97	0.60	8.27
17	3.97	131	1.41	0.00	0.70	2.97	0.47	8.28
18	4.00	125	1.39	0.00	0.70	3.01	0.46	8.31
19	4.18	124	1.39	0.19	0.70	3.20	0.46	8.50
20	4.19	95	1.31	0.20	0.66	3.21	0.44	8.51
21	4.22	93	1.30	0.23	0.65	3.24	0.43	8.54
22	4.40	96	1.32	0.41	0.66	3.42	0.44	8.73
23	4.41	72	1.23	0.42	0.62	3.43	0.41	8.74
24	4.42	76	1.25	0.43	0.62	3.44	0.42	8.75
25	4.44	80	1.26	0.45	0.63	3.46	0.42	8.77
26	4.62	79	1.26	0.63	0.63	3.64	0.42	8.94
27	4.63	64	1.20	0.64	0.60	3.65	0.40	8.95
28	4.65	61	1.19	0.65	0.59	3.66	0.39	8.97
29	4.66	61	1.19	0.66	0.59	3.67	0.39	8.98
30	4.84	58	1.17	0.83	0.58	3.84	0.39	9.14
31	4.85	53	1.15	0.84	0.57	3.85	0.38	9.15
32	4.87	52	1.14	0.85	0.57	3.86	0.38	9.17
33	4.88	52	1.14	0.86	0.57	3.87	0.38	9.18
34	5.06	53	1.15	1.02	0.57	4.03	0.38	9.33
35	5.07	47	1.11	1.03	0.56	4.04	0.37	9.34
36	5.10	49	1.12	1.06	0.56	4.07	0.37	9.37
37	5.28	45	1.10	1.20	0.55	4.21	0.36	9.52
38	5.29	37	1.04	1.21	0.52	4.22	0.35	9.53
39	5.30	37	1.04	1.22	0.52	4.23	0.35	9.54
40	5.32	37	1.04	1.24	0.52	4.24	0.35	9.55
41	5.50	40	1.06	1.38	0.53	4.39	0.35	9.69
42	5.51	38	1.05	1.39	0.52	4.40	0.35	9.70
43	5.53	38	1.05	1.40	0.52	4.41	0.35	9.71
44	5.54	41	1.07	1.41	0.53	4.42	0.36	9.72
45	5.94	64	1.20	1.71	0.60	4.72	0.40	10.03
46	5.95	34	1.02	1.72	0.51	4.73	0.34	10.04
47	5.98	34	1.02	1.75	0.51	4.76	0.34	10.06
48	6.15	29	0.97	1.86	0.49	4.88	0.32	10.18
49	6.16	29	0.97	1.87	0.49	4.89	0.32	10.19
50	6.20	27	0.95	1.90	0.48	4.91	0.31	10.34
51	6.38	26	0.94	2.03	0.47	5.04	0.31	10.63
52	6.82	28	0.96	2.32	0.48	5.33	0.32	10.65
53	6.86	24	0.92	2.34	0.46	5.35	0.30	10.65
54	7.92	32	1.00	2.97	0.50	5.98	0.33	11.28
55	7.96	24	0.92	2.99	0.46	6.00	0.30	11.30
56	8.00	16	0.80	3.01	0.40	6.02	0.27	11.32
57	8.13	14	0.76	3.08	0.38	6.09	0.25	11.39
58	8.14	14	0.76	3.09	0.38	6.10	0.25	11.40

B.2 Optimizing the Rate by a Proper Selection of the Initial Code Word

A non-zero initial code word [icw], can be selected which results in a block code with more number of code words and hence a higher rate. These optimizations for initial code words have been carried out for a few code searches. The table B.55 gives a few cases where a non-zero optimum initial code word is selected which for the same gain gives the maximum number of code words and hence the maximum rate. The percentage increase in the number of code words have been tabulated.

Table B.55: Optimizations for initial code word

Sr. No.	sign. const.	n	$d^2_{\min}$	No. of cd wds zero icw	Optimum icw	No. of cd wds opt icw	Increase in %
1	3-PSK	3	6.0	7	002	9	28.6
2	3-PSK	4	6.0	21	0111	27	28.6
3	3-PSK	5	9.0	9	01121	18	100
4	3-PSK	6	9.0	24	002121	34	41.7
5	3-PSK	7	9.0	72	0002121	85	18.1
6	3-PSK	7	12.0	24	0000022	33	37.5
7	3-PSK	8	9.0	198	00001212	211	6.56
8	3-PSK	8	12.0	60	00111111	99	65
9	3-PSK	8	15.0	18	02201111	23	27.8
10	4-PSK	4	6.0	16	0212	19	18.8
11	4-PSK	5	8.0	32	01203	33	3.1
12	4-PSK	5	10.0	8	10000	12	50
13	4-PSK	6	10.0	16	002223	26	62.5
14	4-PSK	7	10.0	64	3001131	84	31.25
15	5-PSK	4	5.0	20	1122	27	35
16	5-PSK	4	5.52	18	1401	26	44.4
17	5-PSK	4	6.38	14	0244	17	21.4
18	6-PSK	3	5.0	12	033	14	16.7
19	6-PSK	3	6.0	7	004	9	28.6
20	6-PSK	3	7.0	4	122	5	25
21	6-PSK	4	4.0	48	0502	58	20.8
22	6-PSK	4	5.0	42	1522	54	28.6
23	6-PSK	4	6.0	31	0404	33	6.5
24	6-PSK	4	7.0	12	0333	16	33.3
25	7-PSK	3	4.55	12	024	13	8.33
26	7-PSK	3	4.90	10	002	11	10
27	7-PSK	3	3.20	27	104	29	7.4
28	7-PSK	4	3.20	81	1014	87	7.4

B.3 Comparisons of Codes Obtained for Arbitrary Channel Signals Constellations

In this section a comparison between the various arbitrary channel signal constellations is summarized. This provides information for the selection of a channel signal constellation and a coding scheme best suited for a specific application.

In the following tables, for illustration, the criteria selected is that the code rate should be greater than and approximately equal to one in comparison with binary uncoded transmission. Results for various values of block length n, are tabulated.

Table B.56: Comparison between various arbitrary channel signal constellations for $n = 3$

Sr. No.	Sign. const.	rate	CG dB
1	4-PSK	1.00	1.77
2	Asy-4PSK	1.00	0.97
3	5-PSK	1.11	0.97
4	6-PSK	1.19	0.97
5	7-PSK	1.06	1.22
6	8-PSK	1.19	1.76
7	Asy-8PSK	1.00	1.20
8	8-AM PM	1.00	0.00
9	16-QAM	1.00	1.71

Table B.57: Comparison between various arbitrary channel signal constellations for $n = 4$

Sr. No.	Sign. const.	rate	CG dB
1	4-PSK	1.00	3.01
2	Asy-4PSK	1.00	1.77
3	5-PSK	1.04	1.40
4	6-PSK	1.24	1.77
5	7-PSK	1.00	1.80
6	8-PSK	1.15	1.76
7	Asy-8PSK	1.00	1.76
8	8-AM PM	1.04	0.41
9	16-QAM	1.00	2.97

Table B.58: Comparison between various arbitrary channel signal constellations for $n = 5$

Sr. No.	Sign. const.	rate	CG dB
1	4-PSK	1.00	3.01
2	Asy-4PSK	1.00	2.43
3	5-PSK	1.03	2.37
4	6-PSK	1.01	2.43
5	7-PSK	1.00	2.30
6	8-PSK	1.02	2.32
7	Asy-8PSK	1.00	3.01
8	8-AM PM	1.01	0.83
9	16-QAM	1.00	2.97

B.4 Results on Concatenated BCM-BCM Schemes

The tables on **concatenated BCM-BCM schemes**, give some example codes for schemes described in Chapter 7. An actual expanded channel signal constellation of 8-PSK is selected for the examples. Using this for a block length of 3, code words are obtained at various $d^2_{\min}$. Using these code words obtained, as a virtual channel signal constellation, further over a block length of 3, again code words are obtained. Comparisons of these code words obtained

Table B.59: Comparison between various arbitrary channel signal constellations for $n = 6$

Sr. No.	Sign. const.	rate	CG dB
1	4-PSK	1.16	3.01
2	Asy-4PSK	1.00	3.01
3	5-PSK	1.05	2.57
4	6-PSK	1.04	3.01
5	7-PSK	1.00	2.87
6	8-PSK	1.10	3.01
7	Asy-8PSK	1.17	3.01
8	8-AM PM	1.04	1.20

Table B.60: Comparison between various arbitrary signal constellations for $n = 7$

Sr. No.	Sign. const.	rate	CG dB
1	4-PSK	1.29	3.01
2	Asy-4PSK	1.00	3.01
3	5-PSK	1.06	2.88
4	6-PSK	1.10	3.01
5	7-PSK	1.00	3.06

Table B.61: Comparison between various arbitrary channel signal constellations for $n = 8$

Sr. No.	Sign. const.	rate	CG dB
1	4-PSK	1.00	3.97
2	Asy-4PSK	1.00	3.01
3	5-PSK	1.00	3.33
4	6-PSK	1.03	3.52
5	7-PSK	1.05	3.19

with uncoded schemes are tabulated.

Actual expanded channel signal constellation is 8PSK.

Virtual channel signal constellation is 44 point BCM with $d^2_{\min} = 2.59$.

Table B.62: Virtual channel signal constellation is 44 point BCM

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with QPSK rate	Comp. with QPSK CG dB	Comp. with binary rate	Comp. with binary CG dB
1	3.18	20160	0.79	2.01	1.59	-0.99
2	4.00	16120	0.78	3.01	1.55	0.00
3	5.18	8916	0.73	4.13	1.46	1.12
4	6.00	2456	0.63	4.77	1.25	1.76
5	7.18	1336	0.58	5.55	1.15	2.54
6	8.00	761	0.53	6.02	1.06	3.01
7	9.18	427	0.49	6.61	0.97	3.61
8	10.00	240	0.44	6.99	0.88	3.98

Actual expanded channel signal constellation is 8PSK.
Virtual channel signal constellation is 64 point BCM with $d^2_{\min} = 2.00$.

Table B.63: Virtual channel signal constellation is 64 point BCM

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with QPSK rate	Comp. with QPSK Cod. gain	Comp. with binary rate	Comp. with binary Cod. gain
1	4.00	131072	0.94	3.01	1.89	0.00
2	6.00	8192	0.72	4.77	1.44	1.76
3	8.00	4096	0.67	6.02	1.33	3.01
4	10.00	512	0.50	6.99	1.00	3.98

Actual expanded channel signal constellation is 7PSK.
Virtual channel signal constellation is 57 point BCM with $d^2_{\min} = 3.0$.

Table B.64: Virtual channel signal constellation is 57 point BCM

Sr. No.	$d^2_{\min}$	No. of cd wds	Comp. with QPSK rate	Comp. with QPSK Cod. gain	Comp. with binary rate	Comp. with binary Cod. gain
1	3.20	37806	1.07	0.28	1.69	-0.96
2	3.80	32918	1.05	1.03	1.68	-0.22
3	4.50	22140	1.01	1.76	1.60	0.51
4	5.30	3991	0.84	2.47	1.33	1.22
5	6.05	3111	0.81	3.05	1.29	1.80
6	7.00	1469	0.74	3.68	1.17	2.43
7	8.10	680	0.66	4.31	1.05	3.06
8	9.05	411	0.61	4.80	0.96	3.55
9	10.00	247	0.56	5.23	0.88	3.98

Appendix C

Sphere Packings

The primary goal of this appendix is to recollect definitions and results from the theory of sphere packing [21], which are important in the context of block codes as required for this thesis. An introductory article on the packing of spheres [77] is interesting. The exposition in this appendix is rather brief and readers interested in the details may refer Conway and Sloane [21].

Definition C.1 *A system of equal closed spheres in $\tilde{n}$-dimensional space is said to form a packing, if no two spheres of the system have any inner point in common.*

Definition C.2 *If S is an open $\tilde{n}$-dimensional sphere of radius r and content $V_{\tilde{n}}r^{\tilde{n}}$, then,*

$$V_{\tilde{n}} = \frac{\pi^{\frac{1}{2}\tilde{n}}}{\Gamma\left(\frac{1}{2}\tilde{n}+1\right)}.$$

Definition C.3 *A $\tilde{n}$-dimensional packing is said to have the density ρ, if the ratio of the volume of the part of a cube covered by the spheres of the packing to the volume of the whole cube tends to the limit ρ, as the side of the cube tends to infinity.*

Definition C.4 *The center density, is the density divided by $V_{\tilde{n}}$.*

It is the number of centers of unit spheres in the $\tilde{n}$-dimensional space per unit content.

Definition C.5 *The number of spheres touching a given sphere in a packing of $\tilde{n}$-dimensional spheres is known as the contact number or the kissing number,*[1] *denoted by τ, of that sphere.*

[1] Other names for the contact number are Newton number, co-ordination number or legacy.

For a lattice packing τ is the same for every sphere, but for an arbitrary packing τ may vary from one sphere to another.

A131061

A131061